

O

T

S

D

AR-010-111

DSTO-TR-0481

Using Evolutionary Acquisition
For The Procurement Of
Complex Systems

Derek E. Henderson
and Andrew P. Gabb

APPROVED FOR PUBLIC RELEASE

© Commonwealth of Australia

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Using Evolutionary Acquisition for the Procurement of Complex Systems

Derek E. Henderson and Andrew P. Gabb

Information Technology Division
Electronics and Surveillance Research Laboratory

DSTO-TR-0481

DTIC QUALITY INSPECTED 2

ABSTRACT

Technical Report

This report presents the results of a study into the future use of evolutionary acquisition (EA) in the procurement of complex systems for the Australian Defence Organisation. It identifies the benefits, penalties and risks of EA projects, analyses the different activities involved, and makes recommendations for project planning, management and development activities. Its objective is to provide both an understanding of the dynamics of EA projects and guidance for the use of EA in practice.

APPROVED FOR PUBLIC RELEASE

19970724 089

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Published by

*DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury South Australia 5108*

Telephone: (08) 8259-5555

Fax: (08) 8259-6567

© Commonwealth of Australia 1997

AR-010-111

March 1997

APPROVED FOR PUBLIC RELEASE

Using Evolutionary Acquisition for the Procurement of Complex Systems

Executive summary

This report presents the results of a study into the future use of evolutionary acquisition (EA) in the procurement of complex systems for the Australian Defence Organisation. It identifies the benefits, penalties and risks of EA projects, analyses the different activities involved, and makes recommendations for project planning, management and development activities. Its objective is to provide both an understanding of the dynamics of EA projects and guidance for the use of EA in practice.

The study was carried out under task ALO 95/244, *Evolutionary Acquisition Strategies* and sponsored by the First Assistant Secretary Defence Materiel (FASDM). It builds upon the previous studies undertaken by the Defence Acquisition Organisation's Evolutionary Acquisition Tiger Team.

The limited use of EA in the past reflects a concern that EA and similar strategies are inherently risky. However, serious problems in the development of complex systems by traditional strategies have led to the realisation that these strategies also involve significant risks for certain types of projects, risks that may be greater than those in EA strategies. There is now wider acceptance that we need to adopt more responsive and iterative strategies, that we need to accept and understand the risks, and control them.

The main thrust of EA is the specification, design, implementation, testing, delivery, operation and maintenance of systems incrementally. Delivery of each incremental release increases the capability of the system until complete. Users have early access to system releases and are encouraged to provide feedback on performance. This is used to shape the system as it evolves into its final form. Projects which are suited to EA will normally have some or all of the following characteristics:

- Software intensive systems.
- Systems using rapidly changing technology, e.g. computer based systems.
- Systems where humans are an integral part of the system.
- Systems with a large number of diverse users.
- The system is unprecedented.
- A limited capability is needed quickly.

In other words, systems suited to EA will have requirements which are difficult to capture and define and which are likely to change during the life of the project. EA projects are also likely to utilise commercial technology that is emergent or quickly changing, making it particularly relevant for IT projects (projects with a high information technology content).

When used correctly, EA can provide significant benefits for appropriate projects over traditional acquisition models. The benefits are likely to include:

- Better and better understood requirements earlier in the acquisition process.
- Fielding an early operational capability.
- The ability to incorporate new technology.
- More control and visibility of project progress.
- Systems which more closely meet their users' needs when delivered.
- Lower support costs because the need for early modifications is reduced.

These benefits are offset by EA requiring a higher level of resources in both the number and quality of staff. It may also cost more and take longer than a successful traditional strategy because of the interactive and iterative nature of the project. For projects where EA is the best strategy, however, the probability of success using traditional approaches may be quite low. EA also involves higher risks in a number of areas which need to be controlled if the project is to be successful.

This paper suggests techniques for maximising the benefits while identifying and managing risks, and includes discussion of the following areas:

- Planning EA strategies.
- Acquirer resources and staff skills needed.
- Selecting a supplier.
- The importance of the system architecture.
- Interaction between the acquirer, supplier and user staff.
- Determining and evolving the system requirements.
- Allocation of requirements and functions to increments and phases.
- The handling of incremental system releases.
- The role of the users in EA projects.
- Test and evaluation activities.
- Alternative and supplemental activities to EA.

Finally, a section on troubleshooting in EA projects is provided where a number of possible problems in EA projects are examined and analysed, and remedial actions are suggested to get the project back on track.

There may be a tendency to think that EA is an easy option for acquisition, that it simplifies acquisition, or that it reduces the need for early project definition activities. None of these are true. In practice EA requires more planning, more effort and more discipline than traditional acquisition approaches if it is to be successful. We believe however that there are real benefits to be gained from the EA approach, but there are also significant risks. For some projects an iterative approach such as EA is essential.

Authors

Derek E. Henderson

Information Technology Division

Derek E. Henderson has been with the Australian Defence Science and Technology Organisation (DSTO) since 1991. Prior to that he was with the Ministry of Defence (MOD) in the UK after graduating in Electronic Systems Engineering at the Royal Military College of Science, Shrivenham, UK. Since joining DSTO he has undertaken studies for the Australian Department of Defence in the areas of specification, interoperability, and acquisition of information systems. His current research interests include C3I front end systems engineering, evolutionary acquisition and C3I database systems.

Andrew P. Gabb

Information Technology Division

Andrew Gabb has been with the Australian Defence Science and Technology Organisation since graduating in Electrical Engineering and Computing Science at the University of Adelaide in 1971. In that time he has acted as a consultant to numerous major projects in the Australian Department of Defence, in the areas of systems and software engineering primarily for combat systems. He is currently a Principal Research Scientist conducting research into C3I front end systems engineering and procurement methods.

Contents

1. INTRODUCTION	1
1.1 Purpose	1
1.2 Scope	1
1.3 Background	2
1.4 Acknowledgments	3
2. TERMINOLOGY	4
3. OVERVIEW OF EA	6
3.1 The basic EA model	7
3.2 How does EA differ from traditional acquisition models?	8
3.3 What EA is not	9
3.4 Project indicators for EA	10
3.5 Risk indicators for EA	11
4. BENEFITS, PENALTIES AND RISKS IN EA	13
4.1 Introduction	13
4.2 Benefits	13
4.3 Penalties	15
4.3.1 Penalties in implementation cost and schedule	15
4.3.2 Penalties in management resources and activities	16
4.4 Risks	17
4.5 Benefits, penalties and risks in different acquisition models	18
5. JARS - AN EXAMPLE EA PROJECT	19
5.1 An overview of JARS requirements	20
5.2 Applying EA to JARS	21
6. PLANNING AND MANAGING EA PROJECTS	22
6.1 Planning and funding approval	22
6.1.1 Planning for funding approval	22
6.1.2 The use of a steering committee	23
6.1.3 Acquirer resources and staff skills	24
6.1.4 Maintaining a strong negotiating position	25
6.2 Planning an EA strategy	27
6.2.1 Dividing the project into phases and increments	28
6.2.2 Using an architecture design phase	28
6.2.3 Identification of floating increments	29
6.2.4 Fielding of releases	29
6.2.5 Planning system support	30
6.2.6 Planning for the use of modern process standards	31
6.2.7 Documentation requirements	31
6.3 Selecting a supplier	33
6.4 Project management	34
6.4.1 General	34
6.4.2 Interaction between the acquirer, supplier and users	35
6.4.3 Risk management	36
6.4.4 Configuration management	37
7. THE SYSTEM ARCHITECTURE	37
8. REQUIREMENTS AND EA	38
8.1 The form and nature of requirements	38
8.2 The risk of inadequate requirements	38
8.3 Bounding the requirements	40
8.4 Requirements volatility	40
8.5 Requirements capture and definition	41
8.6 Requirements evolution	41
8.7 Requirements, solutions and COTS components	42
8.8 Requirements management	43
9. INCREMENTS AND RELEASES	44
9.1 The nature of increments	44

9.2 Factors affecting increment selection	44
9.3 Floating increments.....	46
9.4 Concurrent development of increments.....	46
9.5 Releases	46
9.5.1 Non-fielding of releases.....	46
9.5.2 Disabling of functions.....	47
9.5.3 The first increments and first releases	47
9.5.4 The initial fielded system.....	47
9.6 Change control during the development of increments	48
10. THE ROLE OF THE USERS	48
11. TEST AND EVALUATION.....	50
11.1 General.....	50
11.2 Acceptance tests.....	51
11.3 Planning test and evaluation	51
11.4 Correcting defects including the use of warranties.....	52
12. ALTERNATIVE AND SUPPLEMENTAL ACTIVITIES TO EA.....	53
12.1 The use of prototypes.....	53
12.2 The use of test beds.....	54
12.3 Pilot systems.....	54
13. TROUBLESHOOTING	55
13.1 Interaction and communication problems	56
13.2 Cost blowout	56
13.3 Schedule slippage	57
13.4 Acquirer decision making is slow	57
13.5 Delays in gaining phase approval.....	57
13.6 Requirements blowout.....	58
13.7 Excessive requirements volatility	58
13.8 Slow requirements evolution	59
13.9 Excessive prices for later phases and increments	59
13.10 Architecture problems	60
13.11 Technical problems	60
13.12 Deferment of functions	60
13.13 Deferment of associated deliverables.....	61
13.14 Non-delivery of agreed functions	61
13.15 Shortfall in quantitative performance.....	61
13.16 Poor reliability of fielded system.....	62
13.17 Conflicts in supplier/user access to the fielded system.....	62
14. CONCLUSIONS	63
15. REFERENCES	63
 Table 1. Comparison of acquisition model characteristics (typical)	 9
Table 2. Penalties, resources and risks for different acquisition models	19
Table 3. JARS phases and increments.....	21
 Figure 1. The basic EA model.....	 8

Abbreviations

ADO	Australian Defence Organisation
AFCEA	Armed Forces Communications and Electronics Association
ALO	Acquisition and Logistics Organisation (now the Defence Acquisition Organisation)
CEPMAN	The Capital Equipment Procurement Manual
COTS	Commercial Off The Shelf
DMD	Defence Materiel Division
DSMC	Defence Systems Management College
DSTO	Defence Science and Technology Organisation
EA	Evolutionary Acquisition
ED	Evolutionary Development
FASDM	First Assistant Secretary (Defence Materiel)
GD	Grand Design
IA	Incremental Acquisition
IP	Intellectual Property
IT	Information Technology
JARS	Job Action and Reporting System
PDS	Preliminary Design Study
RM	Requirements Management
T&E	Test and Evaluation
US	United States

1. INTRODUCTION

1.1 Purpose

This report presents the results of a study into the future use of evolutionary acquisition (EA) in the procurement of complex systems for the Australian Defence Organisation. It identifies the benefits, penalties and risks of EA projects, analyses the different activities involved, and makes recommendations for project planning, management and development activities. Its objective is to provide both an understanding of the dynamics of EA projects and guidance for the use of EA in practice.

The study was carried out under task ALO 95/244, *Evolutionary Acquisition Strategies* and sponsored by the First Assistant Secretary Defence Materiel (FASDM). It builds upon the previous studies undertaken by the Defence Acquisition Organisation's Evolutionary Acquisition Tiger Team.

This study is complementary to the work done by the EA Tiger Team. Having agreed that EA is a valid and necessary approach to the acquisition of complex systems there was a need to examine EA at a more detailed level; to investigate the activities, relationships and problems in EA projects, and to provide some guidance for those considering using EA.

Many of recommendations and suggestions provided here are equally valid for other acquisition strategies. However, because of the dynamic nature of EA projects, they assume a greater significance, and impose a greater risk if they not are taken seriously.

The views expressed in this paper are those of the authors and do not represent the policy or official standpoint of the Australian Department of Defence.

1.2 Scope

This paper addresses the form of evolutionary acquisition which is likely to be used in the Australian Defence Organisation (ADO). It is one way in which evolutionary techniques can be used for the acquisition of complex systems, but is by no means the only way. Because the term 'evolutionary acquisition' can be applied to a number of different acquisition models, both practical and theoretical, we have carefully described the model presented here, in section 3. To some extent, this form of EA is constrained by the project approval policy and processes in the ADO. These constraints and their effects are not discussed here in any detail but can be found in the EA Tiger Team final report [DMD 1996] and the ADO's Capital Equipment Procurement Manual [CEPMAN 1992] described in the next section.

An 'EA project' in this context is an acquisition project for the delivery of a required capability by the means of evolutionary acquisition. It is assumed that there is a common understanding between the users and acquirers on what the

'required capability' is, and that it will be possible to recognise when that capability has been achieved. In other words, EA projects are not open ended, and are not intended to be used for through life system enhancement and support.

Despite these constraints on the EA model, we believe that many of the activities and findings discussed in this paper will also be applicable to other forms of EA.

1.3 Background

Over recent years the advance of Information Technology (IT) and rapid improvement in the performance of computer hardware has led to the availability of computer based, software intensive systems with unprecedented power and range of capabilities. These systems are so complex, and their technologies (including software) are changing so rapidly that the potential users have great difficulty in specifying many of their detailed needs. Traditional acquisition strategies often fail (or are unable) to take this into account - the stated user requirements remaining static after the development contract is signed. Additionally, advances in technology are not easily incorporated into systems when the advances occur during the project development cycle. Consequently, many of the resultant systems do not meet the users' expectations, cost too much and take too long to develop (due to the strategy's weakness in accommodating changes), and cost more to maintain (including enhancements) than to build. EA is an alternative approach with the potential to produce significantly better results.

EA is sometimes portrayed as a new concept. In fact, iterative strategies similar to EA have been used for many years in the development of applications where the needs were not clear at project outset. That EA has not been more widely used reflects the concern of those who approve the acquisition strategies and funding: the concern that EA and similar strategies are inherently risky, particularly with regard to exploitation of the acquirer by the supplier. The difficulty in developing convincing business cases for iterative strategies is also a contributing factor. Failures in recent years in the development of complex systems by traditional strategies (e.g. big bang, incremental or phased acquisition) have led to the painful realisation that these strategies also involve significant risks for certain types of projects, risks that may be greater than those in EA strategies. The implication is clear: to acquire the systems we need, we need to adopt more responsive and iterative strategies, and we need to accept and understand the risks, and control them.

The main thrust of EA is the specification, design, implementation, testing, delivery, operation and maintenance of systems incrementally. Delivery of each incremental release increases the capability of the system until complete. Users have early access to system releases and are encouraged to provide feedback on performance. This is used to shape the system as it evolves into its final form. If this approach is followed in a disciplined manner, a more capable and relevant system should result.

Many of the EA concepts have resulted from work done in the US at the Defence Systems Management College [DSMC 1995]. These and other studies have been closely followed in Australia and a small number of C3I systems are currently under development using acquisition strategies having some EA characteristics. Fairs [1992] provides further discussion about the experience of the Australian Defence Organisation (ADO) with iterative methods, particularly EA.

In 1995 a study was undertaken by the Evolutionary Acquisition Tiger Team, which included staff from wide ranging areas within the ADO, including the authors of this report. The objectives were:

- To study the concepts of evolutionary and iterative acquisition and identify the significant issues involved in its application in Defence.
- To report on potential benefits, costs, areas of application, areas of incompatibility between EA and existing acquisition procedures, and to propose solutions to these.

The outputs of the study were:

- A report recommending that EA be considered for selected projects and proposing models for EA [DMD 1995].
- Draft chapters for the Capital Equipment Procurement Manual [CEPMAN 1992] providing guidance on the use of EA and explaining its relationship to other acquisition models [DMD 1996a and 1996b].

A wide range of stakeholders were consulted during the course of the study, including policy makers, Defence project managers and industry. The study proved successful, with wide agreement that EA and other iterative acquisition strategies are appropriate for the acquisition of some complex systems for the Australian Defence Organisation. EA has since been endorsed for use within the ADO, in the form suggested.

1.4 Acknowledgments

The authors wish to acknowledge the work of the many authors contributing to this field, but particularly that of the US Defence Systems Military College [DSMC 1995], the Evolutionary Acquisition Tiger Team of the Australian Department of Defence [DMD 1995, 1996a and 1996b], Ken Fairs [Fairs 1992], and the AFCEA Evolutionary Acquisition Study Team [AFCEA 1993] who have helped to clarify what was previously a confusing and often misunderstood method of doing business.

2. TERMINOLOGY

Much of the terminology used in this paper is defined in this section. Many of the terms are used commonly in systems engineering and other fields and can have different meanings depending on the context in which they are used. Because of the potential for misinterpretation, this section is arranged in the form of a discussion, rather than an alphabetically arranged list of definitions. This approach allows discussion of meanings and concepts as well as providing definitions.

Many of the definitions in this section are based on those in *System and Software Requirements Engineering* [Thayer and Dorfman 1990]. Some definitions are adapted to reflect the Australian acquisition context and other sources are referenced directly.

Project Sponsor	The office responsible for formulating project policy. The sponsor will normally initiate and justify the project, provide operational requirements and generally represent the users' interests. As such, the sponsor is effectively the 'customer' for the system being acquired.
Acquirer	The acquirer is an organisation that undertakes procurement for itself or another organisation. Often the acquirer is separate from the potential system users (see below). The supplier or developer is an organisation that develops the required products ('develops' may include new development, modification, reuse, re-engineering, maintenance, or any other activity that results in products). The supplier/developer may be a contractor or Government agency. (Adapted from MIL-STD-498.)
Supplier	
Developer	
Users	The users of a system are the operators and supporters of a system, and the trainers that train the operations and support personnel. This definition (from EIA/IS-632) extends the traditional meaning of users to include the support personnel, including trainers. It recognises the fact that the needs of the support personnel must be taken into account in the requirements for an operational system. In this report 'users' also refers to the user representatives who contribute to the project.
Requirement	A requirement is a capability needed by a user to solve a problem or satisfy an objective. An operational requirement is a high level requirement which relates directly to the users' needs in carrying out their mission. It is usually written in the user's language and describes the mission or missions, the operations that need to be performed and the capabilities needed to perform them.
Operational requirement	

System Unprecedented system	A system is a collection of hardware, software, documentation, people, facilities and procedures organised to accomplish a common objective. An unprecedented system is a system for which design examples do not exist so that the physical architecture alternatives are unconstrained by previous system descriptions or implementations.
Architecture	An architecture is the structure of, and relationship between, the components of a system. A system architecture will also include the system's interface with its operational environment.
Acquisition Implementation Life cycle	Acquisition is a method of acquiring systems which includes conceptualisation, design, development, test, contracting, production, deployment, logistic support, modification and disposal of system, supplies or services. Implementation is a subset of acquisition - the process of converting a design into a working system. A system life cycle is all the steps or phases a system passes through from conception to disposal.
Grand design Big bang Traditional approach Waterfall model	The terms grand design and big bang acquisition are often used synonymously. They refer to a traditional method of acquiring systems which is essentially a "once-through, do-each-step-once" strategy, i.e. determine user needs, define requirements, design the system, implement the system, test, fix, and deliver [MIL-STD-498]. Traditional approaches include grand design, incremental acquisition and phased acquisition, where there is a strong emphasis on planning most or all of the project (or phase) in advance. The waterfall model [Royce 1970] is the software model of this process which identifies a number of development activities including analysis, design, coding and testing and stipulates that these activities be done in that order. As with grand design, the waterfall model requires all requirements to be specified at the start of the project and usually assumes that the system will be delivered at the end of the process.
Incremental acquisition	Incremental acquisition involves building a system in a series of increments, each adding to the capability of the previous one. There is usually a single contract and all requirements are assumed to be known at the beginning of the project.
Phased Acquisition	In phased acquisition the acquisition is divided into phases, allowing decision points (and often delays) between phases, when the direction of a project may be significantly changed. The actual acquisition model for each phase is not specified, and might be, for example, grand design or incremental acquisition.

Evolutionary acquisition	Evolutionary acquisition involves the acquisition and early fielding of a well defined initial system with a limited capability followed by a series of enhancements that incorporate planned additional capability as well as improvements, based on feedback from users. This iterative process continues until the required capability is established.
Evolutionary development	Evolutionary development is a <i>development</i> strategy involving the process of progressively adding increased functionality or performance through a series of system builds.
Core system Minimal system Initial fielded system	The core system consists of a system architecture and other elements which provide a basic functionality and common services for the system as a whole. A minimal system consists of the core system plus components providing sufficient functionality to provide a usable capability. The initial fielded system is the first operational system that is delivered, which may, or may not be, the minimal or core system.
Increment	An increment is the basic building block of EA. It is a discrete package of work, agreed by acquirer and supplier, that will normally increase the functionality of the system.
Phase	A phase is a package of work for which funding is separately approved. A phase includes one or more increments.
Release Build	A release or build is a version of a system or system component that incorporates a specified subset of the capabilities. A release will usually be delivered to the acquirer, possibly for operational service, whereas a build need not be delivered.

3. OVERVIEW OF EA

This section introduces the EA model which is assumed throughout this paper. The objective is to provide an understanding of EA principles and to remove any confusion with other models. The basic model is described, EA is compared with other acquisition models and what EA *is not* is discussed. Finally, project characteristics indicating the suitability of EA and the risks associated with these characteristics are described.

The form of evolutionary acquisition described here is that which is likely to be used in the Australian Defence Organisation (ADO). It is one way in which evolutionary techniques can be used for the acquisition of complex systems, but it is by no means the only way.

It should also be stressed that the use of EA is not limited to computer based systems, but the ability to make significant changes to the functionality of such

systems through software changes alone often makes them candidates for EA. For this reason, most of the examples provided are based on computer based systems.

3.1 The basic EA model

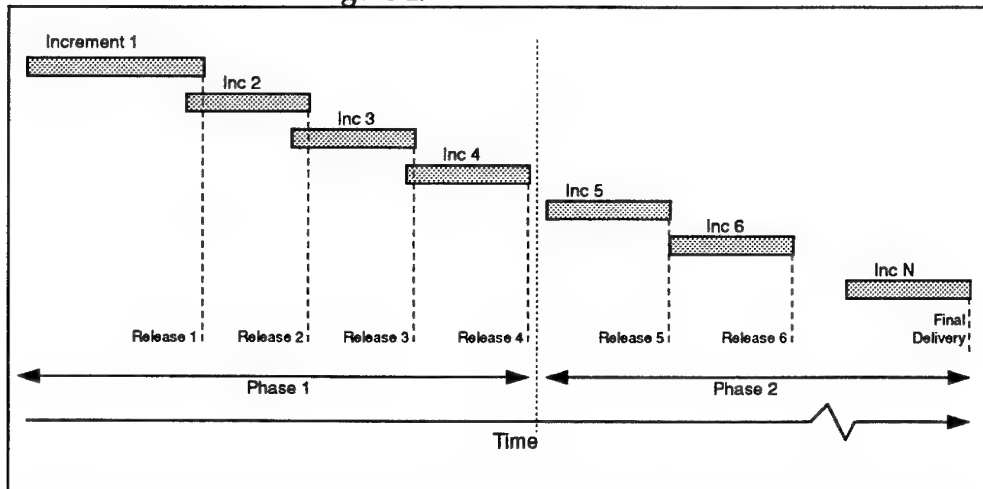
The main thrust of EA is the specification, design, implementation, testing, delivery, operation and maintenance of systems incrementally. Delivery of each incremental release increases the capability of the system until complete. Users have early access to system releases and are encouraged to provide feedback on performance. This is used to shape the system as it evolves into its final form. The aim is to establish an acquisition and development environment which is both sensitive and responsive to the users' needs.

The main features of the basic EA model are as follows. These are discussed in more detail later in this report.

- a. **An incremental approach.** Projects are divided into phases and increments, then developed and acquired incrementally, as shown in Figure 1. Each increment results in the development of functions which increase the overall capability of the system. Each increment may involve a cycle of system development activities from specification through design to testing, fielding and maintenance (see section 9).
- b. **The overall functionality is bounded.** The broad requirements for the system as a whole need to be known prior to the start of design and development (see section 8.3). Despite this, there is a tacit assumption in EA that the requirements may move outside the agreed bounds to a limited extent during the course of the project.
- c. **Requirements.** The detailed requirements for a minimal system and preferably some other early increments are clearly defined at the outset. Other requirements will be progressively refined during the project (see section 8).
- d. **A flexible architecture.** The system architecture needs to support the functions delivered with each release, including those for which detailed requirements have not yet been defined. For this reason the architecture must be flexible, scalable, extensible and maintainable (see section 7).
- e. **Users are part of the process.** The dedicated involvement of users in the development process allows the requirements to be continuously reviewed. Users also contribute by their use and review of early releases of the system, often in an operational environment (see section 10).
- f. **Multiple contracts.** The contract for the first phase will often be for a minimal system, plus additional increments if they can be clearly specified. The capability and price of later increments are agreed prior to

the start of each phase. It is expected, but not assured, that successive phases will use the same supplier (see section 6.1).

Figure 1. The basic EA model



3.2 How does EA differ from traditional acquisition models?

The grand design and incremental acquisition models have been widely used on defence projects, and are relatively well understood. To aid understanding of EA, a comparison of the differences in the main characteristics of EA and these better known models is shown in Table 1.

For the purpose of this comparison brief descriptions of grand design and incremental acquisition follow:

- a. **Grand design.** This has been the most common traditional approach to acquisition (and is also known as the 'big bang' approach). It is essentially a "once-through, do-each-step-once" strategy, i.e. determine user needs, define requirements, design the system, implement the system, test, fix, and deliver. It tends to be a linear process, with completion of an activity being the trigger for commencing the subsequent activity. Contracts are usually awarded competitively covering a single major project, on the basis of complete defined requirement.
- b. **Incremental acquisition.** Incremental acquisition is similar to the grand design approach, except that system design, development and production may occur in a series of increments. Each build may enter operational service when complete, adding to the capability delivered previously. Incremental acquisition is often managed under a single contract. Its main difference from EA is that the requirements, and often the design, tend to be frozen before development begins.

It should be noted that the comparison is between 'typical' projects. It is recognised, for example, that some grand design projects have involved

considerable user involvement. The actual characteristics of acquisition models will vary considerably between different projects.

Table 1. Comparison of acquisition model characteristics (typical)

	Grand design	Incremental acquisition	Evolutionary acquisition
Requirements	Known at project start - remain relatively stable throughout the project.	Known at project start - remain relatively stable throughout the project.	Overall requirements are broadly defined - evolution and refinement of requirements during development.
Design	Determined early.	Determined early.	Architecture design and initial functions determined early - detailed design of other functionality evolves.
Acquisition activities	Acquisition activities occur sequentially.	Acquisition activities occur sequentially.	Many acquisition activities occur repetitively and concurrently.
Delivery	Single delivery.	Incremental delivery.	Incremental delivery.
Contractual arrangements	Competitive tendering. Winner awarded contract for the entire system.	Competitive tendering. Winner awarded contract for the entire system.	Competitive tendering for the first phase only - usually same source subsequently.
Acquisition costs	Assumed known at contract start, and capped.	Assumed known at contract start, and capped.	Assumed known for first phase only - cost for total system estimated and capped.
User involvement	Relatively low.	Relatively low, but some feedback from the use of interim fielded systems.	User involvement is essential and used to shape the requirements for the later increments.

3.3 What EA is not

This section discusses what EA is *not*, at least in the terms of this paper. It has been provided because of the tendency of some writers to use the term 'evolutionary acquisition' very broadly.

Evolutionary acquisition is not necessarily:

- a. **A prototyping approach.** EA does not result in the delivery of prototypes, although prototyping can be a complementary technique (refer to section 12.1). The systems fielded using EA are subject to the appropriate development standards and quality controls during development and have to pass acceptance tests before delivery can occur. Prototypes are not always subject to these rigorous measures and are therefore often not suitable for operational use.
- b. **Evolutionary development (ED).** Evolutionary development is essentially a *development* approach used by a system developer, and is relatively common particularly in software intensive projects. It is also

commonly used in grand design and incremental acquisition projects. An evolutionary acquisition strategy will oblige the supplier to use an evolutionary approach to development.

- c. **A system evolution strategy.** EA is not intended to provide a vehicle for evolving a system throughout that system's lifetime. EA will often be the first step which establishes the system, although many of the techniques discussed in this paper will also be relevant in the upkeep of the system.
- d. **A remedy for poor requirements.** EA is not a substitute for rigorous requirements capture and analysis. It allows the *refinement* of requirements but, like the grand design and incremental acquisition approaches, is vulnerable to poor initial requirements (see section 8).
- e. **Incremental acquisition.** As discussed in the previous section, EA's main difference from incremental acquisition is that the requirements are intended to evolve during development. By comparison, incremental acquisition is much more rigid in its approach.
- f. **Phased acquisition.** Phased acquisition is typically used when there are serious risks, uncertainties or difficulties in planning a project. It provides decision points (and often delays) between phases, when the direction of a project may be significantly changed. There is generally more certainty about the objectives in an EA project, although EA can be viewed as a streamlined phased approach where the delays between phases are eliminated (or at least reduced).
- g. **A strategy to circumvent formal funding procedures.** What appears to be an inherent flexibility in EA is seen by some as an opportunity to gain approval for a project where the objectives and risks are insufficiently defined, and where a phased approach (see above) may be more appropriate.

3.4 Project indicators for EA

EA will not be suitable for every project for a number of reasons. Even where EA may appear to be appropriate, the risks involved may outweigh the benefits, and a more traditional approach will be warranted. Sections 3.5 and 4.4 discuss risk in EA projects.

The indicators below are provided as a guide to when EA *may* be suitable. The final decision on whether to proceed will depend on trading off many different factors as shown in section 4.

Projects which are suited to EA will normally have some or all of the following characteristics. It is no coincidence that each of these is generally regarded as a project risk indicator, regardless of the acquisition strategy used.

- a. **Software intensive systems.** Systems with a large software component are often delivered late, over-budget and do not satisfy user expectations. This has led to what is sometimes referred to as the 'software crisis' [Sommerville 1989].
- b. **Systems using rapidly changing technology such as computer based systems.** Traditional acquisition models often result in the delivery of systems with obsolescent computer hardware, due to the high speed of computer technology advancement. Such systems often require subsequent upgrading at substantial cost.
- c. **Systems where humans are an integral part of the system.** While the performance of hardware or software may be predictable, the effects of the interaction of users with the technology is more difficult to predict. This can make the system's requirements difficult to define. Use of the system will often reveal shortcomings, leading to requests from users for modifications. Similarly, users will see the opportunities for capabilities which were not originally envisaged, leading to requests for enhancements.
- d. **Systems with a large number of diverse users.** Variations in the experience and competence of users makes defining user requirements difficult - particularly in the area of human-computer interface design. This problem is exacerbated when there are a large number of users.
- e. **The system is unprecedented.** Where the system will satisfy an unprecedented requirement, or the potential users of the system have little or no experience with systems of this type, it will be difficult for them to accurately identify their requirements.
- f. **A limited capability is needed quickly.** Operational demands may dictate that a limited capability is required early.

In summary, systems suited to EA will have requirements which are difficult to capture and define and which are likely to change during the life of the project. EA projects are also likely to utilise commercial technology that is emergent or quickly changing, making it particularly relevant for IT projects (projects with a high information technology content).

Additionally, the current changing worldwide strategic defence environment has resulted in less stable operational requirements for a wide range of systems [DSMC 1995]. EA may therefore be relevant for other systems, and apply to some weapon and sensor systems, for example.

3.5 Risk indicators for EA

All of the EA indicators identified in the previous section are also recognised as indicators for risk in complex projects. The result of these risks, particularly

using traditional approaches, is often a serious shortfall in the performance of the resultant systems, sometimes resulting in project failures.

Evolutionary acquisition is an acquisition strategy which attempts to control these risks in a dynamic fashion, i.e. it is a strategy for projects which inherently have significant levels of risk. It is important not to lose sight of this fact. EA presents more challenges in project management, not only because it involves a more dynamic project environment, but because the systems it applies to are more prone to risk in their development. It is therefore not surprising that EA projects require highly capable management skills, and the control of risk is essential to their success.

Under these circumstances, it is important that the decision regarding the use of EA should also consider the risks involved. If the risks cannot be adequately controlled, a traditional approach such as phased acquisition may be a safer, albeit slower, route possibly leading to a less capable outcome.

Indicators of risk are shown below. If these indicators apply, either EA should not be considered, or the problems should be rectified beforehand. Alternate approaches are addressed in section 12.

- a. **Poorly defined requirements.** While EA can compensate to some extent for incorrect or inadequate requirements, it does not replace the need to undertake requirements capture and analysis early in the project. Compensation for poor initial requirements will impact cost, schedule and performance (see section 8).
- b. **Technology chase.** EA should not be used if the current available technology does not support the required capability - EA should use components which are near, but not at, the leading edge of technology [AFSB 1992].
- c. **Technology advances.** If there is low confidence that an architecture can be designed to cater for the expected advances in technology, then it is unlikely that the advances will be able to be accommodated within the scope of the project. A phased acquisition may be more appropriate.
- d. **Immature software development processes.** If the likely suppliers do not have mature software development processes which can accommodate multiple changes to the software, successive changes are likely to reduce the maintainability of the software.
- e. **Limited configuration management skills.** If the likely supplier(s) has limited configuration management skills, there is a serious risk that it will not be possible to meet the increased requirements for configuration management in an EA project (see section 6.4.4).
- f. **Limited acquirer management skills.** If the acquirer cannot provide sufficient project management skills, the acquirer will be unlikely to control the project (see section 6.1.3).

- g. **Limited acquirer resources.** If the acquirer cannot provide the number of staff with the required technical skills (recognising that an EA project requires more and better skilled staff than a traditional project) the acquirer's level of interaction, visibility and control will be seriously degraded (see section 6.1.3).
- h. **User involvement cannot be guaranteed.** If adequate involvement of user representatives cannot be guaranteed, the needed feedback and evolution of requirements may not occur (see section 10).
- i. **Users do not understand EA.** If the users do not understand the principles of EA, the risks will increase and the benefits will decrease (see section 10).
- j. **Extent of workplace impact.** If the proposed project will result in major changes to the environment and the way in which the users will work, there may be resistance from the users in contributing wholeheartedly to the project.

4. BENEFITS, PENALTIES AND RISKS IN EA

4.1 Introduction

This section examines the benefits and risks of EA. Lessons are drawn from a number of examples of EA projects for complex systems [AFCEA 1993; AFSB 1992; Dedrichsen 1990; Erdman 1990; Fairs 1992; Giordano 1991; Ackerman 1995].

Whether a project is a success or a failure is rarely totally clear, and projects which are claimed as a success at some stage by some writers, may later be portrayed as failures by others. Similarly, many successful EA project examples have been a second attempt at a project which has failed using a grand design strategy [AFCEA 1993]. The advantages to any strategy in this situation can be enormous, given the hindsight which was not available to the failed project's management.

4.2 Benefits

When used correctly, EA can provide significant benefits for appropriate projects over traditional acquisition models. The benefits, which are interrelated, can be broadly summarised as follows:

- a. **Better requirements.** Because EA caters for changes in user requirements, there is more scope for incorporating changes during development. The changes in requirement may occur for various reasons including errors or

omissions in the original requirements or genuine changes in the operational need. Early use of the system may also stimulate additional requirements, when the users see what the technology can do for them. EA therefore provides mechanisms whereby feedback can be obtained from users (by a high level of user involvement and early delivery of builds), and mechanisms to incorporate changes in later increments relatively easily. These mechanisms, which also allow for the removal of requirements, should result in higher quality and better validated requirements than in acquisition models which resist change.

- b. **More feasible requirements.** The fielding of early releases of the system helps users begin to understand what is feasible and what is not, resulting in more realistic requirements [Wieringa 1996].
- c. **Better understood requirements.** The continuous involvement of users both in refining requirements for future increments, and in providing feedback on released builds of the system, means that the users are genuinely part of the acquisition process. This not only results in requirements which are continuously validated by the users, but also means that more of the acquirer's and supplier's personnel have an understanding of the actual user requirements.
- d. **Early operational capability.** An important feature of EA is the provision of early releases of the system which may be used operationally, and which are regularly enhanced.
- e. **Incorporation of new technology.** By deferring design and component selection until as late as possible in the project, EA enables the incorporation of more modern equipment and software. This can result in higher performance and lower acquisition costs, and also reduce the through life support costs for the system.
- f. **More control and visibility.** There are various aspects of EA which contribute to a high level of control and visibility by the acquirer. These include:
 - Increased interaction between the acquirer and supplier.
 - The partitioning of development into well defined increments.
 - Release of builds showing clear progress.
 - Testing and validation of progressive builds by the users.

Because of this the acquirer has a clearer view of progress, and risks and latent defects (which may have otherwise been accidentally or deliberately obscured) will often become visible sooner. The number of control points is increased over a traditional acquisition process as well, providing better control of risks. Consequently, decisions to change the direction of a project, or even to terminate it, can be made earlier.

- g. **Better systems.** Because of the flexibility in the EA approach, and the continuous concentration on user requirements, EA should result in better systems than traditional approaches.

4.3 Penalties

The benefits listed in the previous section are not free or automatic. Amongst other things, EA projects are inherently more intensive and difficult to manage than grand design projects. This section lists some of the penalties which will normally be incurred in using EA, as compared with a grand design approach. Note that these penalties will normally be unavoidable and are separate from the possible impact of risks which are discussed in the next section.

Most of these penalties arise from the following sources:

- The additional and duplicated activities associated with the delivery and testing of releases, including the management overhead.
- The additional interaction required between the acquirer and supplier for the agreement of the contents of increments.
- The flexible nature of the project, i.e. the acceptance that changes will occur frequently, leading to changes in direction which may result in nugatory work being done and delays to the schedule.

4.3.1 Penalties in implementation cost and schedule

Comparisons of EA models with more traditional models often give the impression that EA provides a better solution faster and at less cost. While this may be true in some circumstances, it tends to be based on the assumption that a grand design approach, for example, will either fail, or will need additional (originally unplanned) activities which significantly increase its cost and schedule. Too often the comparison is between an actual project failure using a traditional approach and a theoretical EA approach which (it is claimed) would have been successful. Very few comparisons, for obvious reasons, consider successful projects which have used traditional approaches.

An EA approach involves more activities, more changes, and more interaction between participants than a grand design approach, and therefore should cost more to implement and take longer, assuming both projects are successful. It is therefore highly likely that the *projected* cost and schedule of an EA project should be somewhat greater than that for an equivalent grand design project. Whether the *actual* cost and schedule are likely to be greater will depend on the probability of success of using a grand design approach (and the subsequent need for remedial activities) and the effectiveness of project management in each case. For projects where EA is the best strategy, however, the probability of success using other approaches may be quite low.

4.3.2 Penalties in management resources and activities

The penalties include the following:

- a. **More management resources.** An EA project requires close control by both the acquirer and supplier throughout the project. This approach is quite different from using a grand design model for a turnkey system, particularly for the acquirer. The acquirer needs to have much more visibility of progress, and there will be more discussion, negotiation and planning activities (see section 6). EA is not a 'hands off' acquisition model.
- b. **Better management.** Because of the intensive and dynamic nature of EA projects, the acquirer in particular requires higher levels of skills and experience than in traditional projects (see section 6.1.3).
- c. **Impact of concurrent activities.** In a typical EA project, activities such as development, test, operations and support will occur concurrently, with interactions between staff working on different activities. This requires greater planning and coordination than for projects where these activities occur separately or sequentially, and is likely to result in some quite challenging management problems (see section 6.4). In IT projects the issue of organisational change management may be particularly important, and will be complicated by the fielding of multiple releases of the system.
- d. **Better configuration management.** At most stages in an EA project, at least three builds will be under consideration. These will include the most recently released build, the build currently under development, and the build being negotiated for the next increment. High quality configuration management is essential in this environment. Unlike more traditional models, the acquirer must also take an active part in the configuration management of the requirements, because each build will usually correspond to a different functional baseline (see section 6.4.4).
- e. **Technical support for the acquirer.** EA requires more and a higher level of technical support for the acquirer. This need arises because many of the decisions which need to be made by the acquirer and supplier together are based on a tradeoff between operational, technical and development issues (see section 6.1.3). It should be noted that for EA the acquirer needs an understanding not only of the requirements and technical issues, but also of the development process.
- f. **User resources and coordination.** The success of EA depends on the continuous review and feedback from user representatives. Finding the right users, ensuring their availability, and coordinating their input to the project is not a simple task (see section 10).
- g. **Support for fielded releases.** Because multiple releases will usually be fielded in an EA project, often with different characteristics, the effort

required to support these releases (including the training of operational and support staff) will be greater than for a single delivery at the end of the project.

4.4 Risks

EA is a highly dynamic and flexible approach to the acquisition of complex systems. As such, it involves higher levels of risks than more traditional models. The risks involved include the following:

- a. **Requirements risk.** Caused by a temptation to defer requirements definition (see section 8).
- b. **Management risk.** EA is more difficult to control and needs a higher level of skill and experience in management by both the acquirer and supplier. For software intensive projects, Downward [1994] points out that for the supplier grand design is the easiest model to accommodate, and that the evolutionary model could be very difficult, and needs the highest level of systems engineering skills. The need for the acquirer and supplier to maintain a close working relationship with a high level of cooperation is also more critical in EA projects.
- c. **Approval risk.** By adopting a phased approach, EA is vulnerable to delays in funding approval which will increase the schedule, and are also likely to be detrimental to performance and cost [Erdman 1990; Fairs 1992]. Such delays may break the flow of development resulting in the loss of key development staff, and additional costs in winding development down and winding up again (see section 6).
- d. **Architectural risk.** The architecture design is critical to success of the project. If the initial architecture cannot accommodate changes later in the project, the project may fail (see section 7).
- e. **Encouraging change.** By accommodating change, EA may be seen to encourage change. This may result in attempts at excessive optimisation, or the incorporation of changes which are not cost effective.
- f. **Resistance to change.** There will always be some resistance to change by the acquirer and the supplier. The supplier in particular will prefer to freeze the configuration as early as possible, to stabilise the design environment.
- g. **User commitment.** EA depends on committed and involved users, both in requirements definition and assessment of delivered releases (see section 10).
- h. **Direct user involvement.** More user involvement means the users will have more exposure to the design and early builds. Different users will have different opinions on requirements and acceptable solutions, and

may require conflicting or insular views. The user involvement must be coordinated and controlled (see section 10).

- i. **Short term benefits.** While one of the advantages of EA is the acquisition of an early (albeit limited) capability, which may be used operationally, there is a risk of the project becoming driven by the short term operational needs of the users [Wieringa 1996].
- j. **Risk avoidance.** The tendency for suppliers to defer implementation of the more difficult (and more risky) features until a later increment must be curbed.
- k. **Supplier selection risk.** Contractors need to be selected not only for their development ability, but also on the basis of their willingness to work closely and cooperatively with the acquirer (see section 6.3). This may be difficult to assess.
- l. **Exploitation risk.** EA may reduce the bargaining power of the acquirer, because the initial contract will not normally encompass the entire task, and subsequent contracts are unlikely to be competed. In negotiating later increments and phases, the supplier may decide to exploit this situation by quoting unreasonably high prices (see section 6.1.4).
- m. **'Patchwork quilt' effects.** Poorly controlled software changes can severely reduce the quality of software. Similarly, the incremental development of requirements may result in a lack of coherence in the requirements as a whole [Wieringa 1996].

4.5 Benefits, penalties and risks in different acquisition models

Table 2 compares the benefits, resource penalties and risks in EA, grand design (GD) and incremental acquisition (IA) models. It is important to note that the ratings are indicative only, and will vary from project to project. The purpose of the table is to give a broad overview of the advantages of the different models, and a basis for comparison, rather than to claim that any model is clearly superior in any instance. For the GD and IA projects, 'ideal' projects are assumed, i.e. the projects perform more or less as planned, and are rated as successful.

In choosing an acquisition model, it may be useful to consider combinations of the different models that are available. For example, grand design projects often include some early delivery of system components. Similarly, in a large grand design project, EA may be considered for the acquisition of complex subsystems, e.g. the development of a command team trainer for a combat system.

Table 2. Penalties, resources and risks for different acquisition models

Issue	GD	IA	EA	Comments on how EA is likely to perform
BENEFITS				
Performance	•	•	●	EA may provide a higher overall performance,
Project Cost	●	•	•	at a higher project cost (but only compared with 'ideal' GD and IA projects),
Schedule	●	•	•	with a longer time to completion (but only compared with 'ideal' GD and IA projects),
Requirements	•	•	●	giving a better quality & understanding of requirements, and responsiveness to requirements change,
Early capability	•	●	●	allowing an early capability to be delivered to the users.
Technology	•	•	●	EA caters for changing technology,
Control	•	•	●	and provides better opportunities for project control and visibility.
RESOURCE ADVANTAGES				
User involvement	●	•	•	EA needs a higher level of user involvement and commitment,
Acquirer resources	●	•	•	more and better trained/experienced acquirer staff, including technical staff,
Supplier resources	●	•	•	and the supplier also needs more and better trained/experienced staff.
RISK CONTROL				
Management	●	•	•	EA is more difficult to manage,
Performance	•	•	●	but the final system is more likely to meet the users' needs,
Changes	•	•	●	and the risks of requirements changes causing problems are lower.
Architecture	●	•	•	The risk caused by an inadequate architecture is higher in EA,
Technology	•	•	●	but advances in technology are less likely to cause problems.
Approval delays	●	●	•	EA is more vulnerable to delays in phase and increment approval,
Contractor selection	●	●	•	is likely to be more sensitive to choosing the right supplier,
Exploitation	●	●	•	and potentially has a higher risk of exploitation of the acquirer.
Notes: 1. Larger dots indicate higher benefit, lower resources or lower risk, i.e. 'better'. 2. The GD and IA models assume 'ideal' (successful) projects. 3. Ratings are indicative only, and will not apply in all projects.				

5. JARS - AN EXAMPLE EA PROJECT

The following example will be used throughout this document. Although relatively simple, it contains many of the attributes of a system for which EA is applicable. It is also equally applicable to military and commercial environments, which makes it understandable to a wider audience. It should be noted, however, that the size and complexity of the task is deliberately exaggerated. It is highly unlikely, for example, that such a project would need the number of increments suggested, or that such an approach would be effective, because of the relatively small amount of work required for each increment. However, a more complex project would take much longer to describe and be more difficult to understand.

5.1 An overview of JARS requirements

JARS is a Job Action and Reporting System which is intended to assist in the management of 'jobs' within a maintenance organisation. It will replace a manual paper based system which uses a file for each new job, and uses forms for job description, work allocation, actions assigned to staff and time recording. Jobs range considerably in size from small one day jobs, to large jobs lasting a year or more. The delivered system will include a network of workstations which will be distributed throughout the organisation.

The main objectives of JARS are to automate, integrate and simplify job management, and to provide a reliable record of events. The potential users of JARS are about to adopt a new quality management system, certified to ISO 9001, and see JARS as critical in meeting the reporting requirements of the quality standard. They would like a capability as soon as possible, even with limited functionality, so that they can begin defining their new management processes.

The basic requirements include the following:

- The ability to create new jobs, modify job information and close (sign off) jobs.
- The ability to access current job status information.
- Access mechanisms which control the ability of different users to view or change certain information.
- The ability to generate 'actions' on jobs, and assign them to staff.
- A timesheet facility so that staff can allocate their time to actions and jobs.
- A costing facility to identify costs which are attributed to jobs.
- Archiving and other systems management functions to ensure the integrity of the database, and to recover data after system failures.
- Basic reporting functions.

The above requirements are really little more than automation of the previous manual system, with some additional functionality to correct perceived deficiencies. In addition, the users have identified the additional 'enhanced' requirements:

- Job planning tools so that planning staff can use current staff allocations and other information.
- 'Super jobs' which will comprise a number of related jobs.
- Continuous calculation of job costs to date, derived from cost records and timesheets.
- Amalgamation of job timesheets and attendance diary information - these were previously completed on different forms.
- Advanced reporting functions such as Action Reports which show outstanding actions for each staff member.

The detailed requirements for these have not been completely defined, although some aspects have been (e.g. the definition of 'super jobs' and how they will be handled). In general, however, these requirements are sufficiently detailed so that all parties have a reasonable understanding of what functionality is expected by the users.

5.2 Applying EA to JARS

The project is divided into 2 phases, primarily because of uncertainty of the detailed requirements for the second phase. There is also a recognition that the Phase 2 requirements are likely to change, and that additional requirements may be identified during the first phase and as the users gain experience with the system. The phases and increments are planned as shown in Table 3.

Notes on Table 3:

- The description of the functions are highly simplified.
- While incremental builds are based on the requirements, some requirements (e.g. reporting) may only be partly met and completed in a later increment.
- The increments have been planned to (a) provide the users with the required early functionality, and (b) defer functions which are less well defined or whose requirements are most likely to change.
- The architecture must be able to accommodate requirements for Phase 2 as well as Phase 1, so that the software architecture must cater for super jobs, for example.
- Fielding of releases will be preceded by acceptance testing, operational evaluation and some training of staff. Each build which is fielded will need to include conversion of existing data, if applicable.

Table 3. JARS phases and increments

Phase	Inc	Functions	Comments
Phase 1	1-1	Architecture (software and hardware) and core functions; Job functions; Access mechanisms; Limited reporting; Limited system management functions; Limited on-line help.	Delivery includes hardware and network installation.
	1-2	Actions; Timesheets; Improved reporting.	
	1-3	Costing facility; Full system management functions.	
	1-4	Full basic reporting; Full on-line help.	
Phase 2	2-1	Super jobs; Limited Job planning tools.	
	2-2	Continuous cost calculation; Amalgamate timesheets and diaries.	
	2-3	Job planning tools.	
	2-4	Advanced reports.	

As Phase 1 progresses and the initial system is installed, feedback will be provided with regard to defects and requested enhancements. The correction of

defects and the implementation of enhancements (agreed between the users and acquirer) will be negotiated with the supplier, and be included in a later increment, in either Phase 1 or 2. Some enhancements, which may be of lower priority and/or beyond the scope of the original requirements, may be deferred to be included in later upgrades to the system, after the completion of the project.

Tenderers for JARS were required to bid for the entire project, with costing of different phases clearly separated. After negotiation, the initial contract was awarded for Phase 1 only, with an agreement that the acquirer could proceed with Phase 2 at the acquirer's discretion, at the price agreed. The functionality and price agreed for Phase 2 would then be used as a basis for negotiation of this phase.

As part of the contract, the supplier agreed to provide prototypes of workstations early in increment 1-1 to allow users to contribute to the user interface requirements. The basis for the user/supplier/acquirer interaction was clearly defined in the contract, refined soon after contract award, and reviewed regularly throughout development. This included the obligation of the acquirer to make experienced users available, the process of examining user requests which might be seen to be outside the scope of the agreed requirements, and mechanisms for conflict resolution.

6. PLANNING AND MANAGING EA PROJECTS

6.1 Planning and funding approval

6.1.1 Planning for funding approval

Before a project can achieve funding approval it will normally be necessary to demonstrate that:

- The capability to be acquired is clearly defined;
- The maximum cost of acquiring this capability can be estimated;
- A clear and achievable acquisition strategy has been developed; and
- The likely risks have been predicted and can be controlled.

Funding authorities require evidence of careful project planning when considering new projects. The planning will need to include evidence that:

- a. **The acquisition strategy is achievable in terms of the timing between events.** EA projects may have far more milestones than traditional projects, due to multiple increments and phases. The plan should demonstrate that schedules are achievable and propose strategies for dealing with slippages.

- b. **The deliverables of each phase are commensurate with what is needed to proceed with subsequent phases.** It will be necessary to allocate requirements to phases or increments in a manner which is technically feasible and which leads to an evolutionary development of the system (see section 9). The plan should show how a logical evolution of the system will occur.
- c. **Risks reduction occurs early.** Where possible, planning should ensure that known risks are reduced early in the project, so that future planning can be soundly based. Early risk reduction should also reduce the impact of the risks, allowing project adjustment or redirection at a time when the majority of the funding may not have been committed.
- d. **The acquirer's negotiating position can be maintained.** One of the perceived risks of EA is that of the acquirer becoming 'locked-in' to a single supplier without a clear definition of what needs to be done. The acquirer will need to show how value for money can be achieved in the later phases, demonstrating that an adequate negotiating position can be maintained.
- e. **The acquirer will have visibility of progress and risks, and the ability to exert control if needed.** An acquirer's ability to ensure that a project stays on track will depend on the acquirer's visibility of progress and risks, having the appropriate numbers and quality of staff to understand them, and having the contractual right to intervene when necessary. The plan must show that all of these issues are satisfied.
- f. **The project phasing permits flexibility to advance or defer elements of the project according to prevailing conditions.** This is one area where EA should provide a significant advantage over traditional approaches. The flexibility of EA will allow a certain amount of leeway in the scheduling of increments and phases.

Addressing these aspects of the plan should help to determine if EA is an appropriate acquisition strategy for the project. For example if the project cannot be split into phases and increments for whatever reason then clearly an alternative strategy should be used.

6.1.2 The use of a steering committee

Delays in the funding approval of phases could put an EA project at risk (see section 4.4). On the other hand, it is inevitable that major funding decisions will be made by an authority independent of the project. This is necessary to ensure that funding decisions are made in a wider context, and also to ensure that the acquirer does not lose sight of the overall project objectives. In an EA project the overall capability is defined, but there is a perceived risk that the project will diverge from its initial objectives, and that the acquirer may lose control of the project's costs and schedule. This risk is understandable considering the management skills needed to run a successful EA project, and the dynamic and flexible nature of EA.

One method of meeting the need for independent approval, and at the same time reducing decision delays is the establishment of a project steering committee, independent of the Project Office, and with the ability to approve funding delegated from the higher authority.

The steering committee may also need to instigate audits of the project to determine that the project is on track. These should include capability audits, where the achieved capability is compared with the projected and approved capability. (This is actually a form of independent validation - independent of both the acquirer and supplier.) In some cases it may also be useful for the steering committee to seek independent specialist advice on technical aspects of the project. Regular independent management audits are also necessary, to ensure that both the acquirer's project office and the supplier are aware of and are controlling project risks.

6.1.3 Acquirer resources and staff skills

Competent staff are needed for effective project management. In EA projects, which are essentially more dynamic than equivalent traditional projects, the need for experienced and skilled staff may be much higher, and the project office needs to be established earlier [DSMC 1995]. The actual staff characteristics needed will depend on several factors including the type of system to be acquired and its complexity, the acquisition strategy and the perceived risks.

Marciniak and Reifer [1990] identify a set of skills which should be considered for any project office:

- Project management
- Engineering
- Configuration management
- Test and evaluation
- Training
- Quality assurance
- Contract administration
- Program control
- Data management
- Interface management
- Logistics support

In EA projects the following systems engineering skills will be particularly important and special efforts may be needed to find staff with the appropriate skills:

- a. **Requirements engineering.** Requirements capture, analysis, definition, validation and management will be ongoing tasks which need dedicated staff. An ability for these staff to work closely with users will be essential.

- b. **Systems analysis.** For complex systems, one or more systems analysts are needed to maintain a systems view, and to assess the impact of changes on the system as a whole.
- c. **System architecting.** Staff with system design skills, and experience in architectures, interfaces, protocols and topologies will be required to assess the merits of architecture proposals, to gauge the impact of changes on the architecture, and to ensure that external system interfaces remain valid.

Additional knowledge required by many of the acquirer staff will include:

- A good understanding of the operational environment and requirement.
- Trends and developments in relevant technologies.
- An understanding of the development process and environment used by the supplier.
- Organisational change management and/or business process reengineering (for certain projects).

To maintain a close and effective working relationship with the supplier, it is also important that the acquirer's staff have experience at working in teams, and have good negotiation and communication skills. Combining training in these skills with the supplier will be beneficial in most cases (see section 6.4.2).

6.1.4 Maintaining a strong negotiating position

6.1.4.1 General

The acquirer will wish to maintain a strong negotiating position in the project to ensure that the result meets the users' needs. The acquirer will also need to demonstrate that this is possible in order to obtain funding approval (see section 6.1.1). This section discusses some of the measures which might be taken to improve and maintain the negotiating position of the acquirer.

6.1.4.2 Possible measures for maintaining an edge

Some of the measures for maintaining a reasonable negotiating position are as follows:

- a. **Source selection on the basis of past performance.** While using past performance as a factor in source selection is aimed at getting the right supplier for a project, it will also affect the attitude of suppliers in all EA projects, if applied uniformly, accurately and decisively (see section 6.3). In fact, this may be the most important factor in the acquirer maintaining an edge in EA projects.
- b. **Understanding value.** In negotiating future phases and increments, the acquirer and users need to have an understanding of the *value* of proposed features. This will allow them to estimate and prioritise different features on the basis of value for money, and will help to focus

the supplier on the same goals. To achieve this, concurrent studies may be necessary to consider the priority of requirements, and the effectiveness and efficiency of alternative solutions.

- c. **Understanding costs.** The acquirer needs to understand the impact of proposed solutions both in terms of the technical issues and the development required, and the basis on which the supplier's cost estimates are established. This requires the acquirer to have more technical support than for traditional projects.
- d. **Understanding the supplier's capability and problems.** The acquirer needs to understand the strengths and weaknesses of the supplier, and problems that the supplier may have in meeting milestones. This will both reduce the likelihood of unrealistic demands being placed on the supplier (and hence milestones not being met), and also allow the acquirer to negotiate with the knowledge of what is achievable and what is not.
- e. **Avoiding intellectual property barriers.** To enable the acquirer to assign future work to another supplier, to change suppliers, or support the system, the acquirer will normally need certain future rights to the intellectual property (IP) developed under the contract. The rights should include as a minimum the rights needed for the acquirer to continue development, and the rights to assign similar rights to a third party. Initial planning, bid evaluation and negotiations need to recognise the risks of including proprietary material in the system, and the use of proprietary tools in development, to ensure that they do not unduly reduce the negotiating position.
- f. **Requiring appropriate development methods and standards.** If the broad development methods and standards are unusual, obscure, or poorly defined, it will usually be extremely difficult for another supplier to contribute to, maintain or take over the development of the system.
- g. **Ensuring methods and standards are followed.** If appropriate development methods and standards are not followed, it will be both difficult and expensive for any supplier (including the proposed supplier) to contribute to, maintain or take over the development of the system. It is important for the acquirer to ensure that the management processes are in place to support the development processes, and to ensure that the stated processes are in fact being followed during the project.
- h. **The identification of optional work.** Early identification of work which may be undertaken by other suppliers, and which is not in the initial contract, provides the acquirer with some flexibility and leverage (section 6.2.3 refers to the identification of floating increments, for example).
- i. **Incentives.** The use of incentive payments (i.e. additional payments for exceptional performance under the contract) might also be considered, although there are some indications that such devices may in fact put undue pressure on the acquirer/supplier relationship.

6.1.4.3 Changing the supplier - a desperate option

While dividing the project into phases will provide the acquirer with the opportunity to change the supplier, this course of action will rarely be seen as cost effective. One reason for this is the unavoidable delay as a new supplier is chosen and becomes conversant with the system development; another is the difficulty in performing a cost tradeoff between changing the supplier and continuing with the current supplier. There are also likely to be serious problems with the transfer of intellectual property (IP), particularly relating to the design and development environment, and much of the needed IP may not be documented or in a usable form. Ideally at least, if the same team is maintained for the entire project then the expertise accumulated can result in the cost reducing for successive phases [Shore 1990].

For this reason, this report generally assumes that the same supplier will be used for all phases, but also provides advice on how to reduce the impact of changing the supplier. It also emphasises the importance of selecting a supplier who has the capability and commitment to undertake the whole project, i.e. all phases.

However, it will sometimes be necessary to change supplier, possibly due to poor performance, excessive prices or the commercial failure of the supplier organisation, and the acquirer should normally preserve this right in the contract, and plan to reduce the impact of such a change.

6.1.4.4 Alternatives to changing the supplier

Possible alternatives to changing the supplier are as follows:

- a. **Deferring functionality.** The implementation of less critical functions may be deferred beyond the scope of the project (possibly to the support phase).
- b. **Removing requirements.** Less critical requirements may be removed completely, to reduce the cost and schedule of the project.
- c. **Contraction of the current supplier's responsibility.** This option involves reducing the scope of future phases with the current supplier to the absolute minimum which is essential (i.e. cost effective) for this supplier to achieve, and competing the remaining capability.

6.2 Planning an EA strategy

An EA project must have a clearly defined strategy, based on a comprehensive risk assessment, that all parties have access to and understand. Both the strategy and the risk assessment must be frequently reviewed and updated [AFSB 1992].

6.2.1 Dividing the project into phases and increments

The number, content and size of phases should be decided on the basis of risk, and the capability which is required by the users of the system. Generally, capabilities which are well known and specified can be included in the first phase (or early phases). Capabilities which are less well defined should be specified in the later phases. The first system to be fielded must also be operationally useful so that the users can gain real benefit from it.

If there is a likelihood of delays in funding approval then this should also be considered in the planning of phases. This will also suggest the minimisation of the number of phases, and consideration of the ordering of increments within phases so that development may proceed while decisions regarding the next phase are being made.

Other factors are relevant to the planning of both phases and increments and are discussed in detail in section 9. Increment planning should include consideration of the testing needed (see section 11) and whether or not specific releases will be fielded or not (see section 9.5).

Generally, the initial contract (phase) should include as much functionality as can be clearly and safely specified. This approach reduces the acquirer's exposure to contractual risk by achieving a fixed price for a large proportion of the system. It also reduces the likelihood of tenderers 'buying in' to the development, i.e. bidding a low price in the expectation of recouping any losses later in the project.

The schedule proposed must be realistic. EA projects are not easy to manage and an unrealistically optimistic timescale will put undue pressures on both the supplier and acquirer management, which will increase the risk of failure. Alternatively, resisting these pressures will normally result in slip in the project schedule, which may be perceived as a management failure in any case. In estimating the timescale, it is important to include consideration of the needs for interaction and decision making, which may be much higher than in traditional projects. In general, these factors will lead to a schedule which is longer than for an equivalent traditional project.

6.2.2 Using an architecture design phase

Two of the more difficult problems in an EA project are achieving an architecture design which is appropriate for the proposed system and which will accommodate future changes (see section 7), and selecting the preferred supplier (see section 6.2.5). One approach which can reduce both of these risks is planning the first phase of the project as a competitive architecture design phase. This is usually referred to in the Australian Defence Organisation as a funded project definition study (PDS).

Marciniak & Reifer [1990] refer to this approach as two phase acquisition, recommending it for systems with high requirements volatility. They suggest selecting two or more suppliers in the first phase to refine requirements,

develop system concepts and in some cases to begin the design. After the competitive first phase a single supplier may be selected to conduct the full scale development. Fairs [1992], the Defence Systems Management College report [DSMC 1995] and the Defence Industry Study Course [DISC 1995] also recommend the consideration of this technique for EA projects.

The two or more tenderers who participate in this phase would be contracted to provide an architecture design, preferably accompanied by prototypes or models, demonstrating that the architecture is appropriate for the project as a whole. This would include showing the architecture's flexibility in accommodating projected possible changes. In addition, the phase could also be used to propose solutions in other areas which may be seen as risk drivers, such as the user interface philosophy and the support approach.

Two disadvantages of this strategy are the additional cost and the delay (due to the need for a second bid evaluation and source selection) that is incurred. If the delay is too long, advances in technology may undermine the results of the first phase. However, there are significant advantages in risk reduction in this approach, which may reduce costs and save delays later in the project.

6.2.3 Identification of floating increments

In some projects it may be possible to identify sizeable packages of work which can be isolated from the mainstream development of the system, which are relatively low risk and low priority, and where there is reasonable flexibility in when they need to be performed. These may be identified as 'floating increments' (see section 9.3) and used to maintain flexibility and control in the project.

Possible uses of floating increments are as follows:

- Where the work may be done by a different supplier, floating increments may be separately sourced. This may allow work to proceed in parallel if there are schedule slippages, using a different supplier.
- Floating increments may also be used in this way to preserve some degree of competition.
- Floating increments could be used to provide 'interim' work for the supplier during approval delays.

If not needed for other purposes, floating increments would normally be implemented relatively late in the project, after the implementation and delivery of higher priority capabilities.

6.2.4 Fielding of releases

Fielding of new releases has to be planned in such a way as to cause the most benefit and least disruption to users. Note that it will rarely be necessary or useful to field all releases (see section 9.5). Issues which might be addressed include:

- a. **Operational testing of the release.** A release will normally undergo operational testing prior to being fielded (see section 11). In some cases, the release may be deemed unsatisfactory and planning should consider this possibility.
- b. **Staged introduction of the release.** Where there are a number of user sites, it may be installed at only one site initially. If performance is satisfactory, the release may then be installed at other sites. (This is a variation of the use of a pilot site, which is discussed further in section 12.3.)
- c. **Planning for failure.** Because the level of operational testing will rarely be exhaustive, the possibility of serious defects in any release, leading to rejection of the system, needs to be considered and planned for. Gilb [1988] suggests that there should be a planned and easy retreat path from each significant step taken. So, for example, after fielding a release and finding errors it may be necessary to revert to the previously fielded system, possibly with 'backwards' conversion of any data created or modified with the new release. An alternative or complementary strategy might be the correction of the system in the field.
- d. **Rights and responsibilities.** There are several other issues relating to the fielding of releases which need to be resolved early in the project. These include:
 - Who owns the fielded system?
 - Who is responsible for the repair of damage to the system equipment?
 - What rights of access does the supplier have to monitor performance or maintain the system?
 - What are the rights and responsibilities of the site support staff with regard to the system?
 - Who will determine the training needed for each release, and who will provide the training?
 - How will user comments be reported to the supplier and/or acquirer?

6.2.5 Planning system support

It may be difficult to determine the system support requirements at an early stage in an EA project because often the final configuration will not be sufficiently predictable until late in the project. Complicating this problem is the likelihood that system support, including training, maintenance and the provision of support personnel, may need to begin with the initial fielded system. Under these circumstances the support functions will need to be determined early, and evolve with the system. Management of system support will also be more challenging because of the number of different participants and stakeholders involved.

While this approach is likely to increase the effort required for support planning during the project, it should result in a much better understanding of the support requirements for the final system, because of feedback gained from support of the interim releases. In addition, the feedback can also be used to influence design decisions in later increments, to enhance the supportability of the system. The benefits are likely to include more effective support for the system and significant savings in through life support.

It should be noted that these benefits should be achievable regardless of whether the support is provided by the supplier, the user organisation or another contractor, and whether there is a transition of the support agency during the project or after delivery.

6.2.6 Planning for the use of modern process standards

Modern system and software engineering process standards, such as EIA/IS-632 [1994] for systems engineering and MIL-STD-498 (and its derivatives) for software development, place emphasis on informal discussions between the acquirer and supplier on status, risks, approaches and other issues. This contrasts with earlier military standards which are generally much more prescriptive about the roles and responsibilities of the acquirer and supplier.

Where these standards are used (and they will be used in many projects for defence systems) the acquirer will no longer have the apparent protection of a prescriptive standard, requiring a better understanding of the standards and higher levels of management skills.

6.2.7 Documentation requirements

Uninformed, prescriptive and untailored use of documentation standards has often led to much unnecessary effort in projects for military systems [Gabb & Henderson, 1995a]. In EA projects, this problem is likely to be much worse because of the iterative nature of the development, and the number of system releases. In theory at least, 10 increments may mean 10 deliveries, each of which may require a full set of development documentation. This approach will result in a much higher cost, a longer schedule and much more documentation than is likely to be needed. It is evident that documentation is a very serious issue in EA projects.

For this reason it is essential that the documentation requirements are kept to the bare minimum needed to protect the acquirer's interests. This applies to how much documentation is developed, to what standard, in what form and how much is actually delivered to the acquirer.

The acquirer needs to consider the precise needs for documentation, i.e. why visibility and delivery of documentation is required, and why documentation needs to be developed to established standards. Common reasons include the following:

- Insufficient or incorrect documentation, or documentation developed to obscure or inappropriate standards, will reduce the possibility of another supplier competing for subsequent phases, or maintaining the system.
- Correct documentation provides a clear indication of what the system will do, and what it consists of (in terms of design and production).
- Documentation often forms the basis of specification, design and test reviews.
- The availability of documentation can be a reliable indicator of progress.
- The final documentation needs to be of adequate quality to support efficient enhancement and maintenance of the system.

Having decided the needs for documentation, the acquirer should determine how these needs may be met cost effectively. In the early stages of project definition, it will be advisable to discuss this issue with several potential suppliers, so that requirements can be stated clearly in the Request for Tender. Once a supplier is selected, the documentation requirements should be discussed and the approach agreed between the acquirer, supplier, users, supporters, verification and validation (V&V) agents and other relevant parties.

The following measures, used separately or in combination, may be useful in reducing the documentation cost and effort:

- Agreeing delivery of documentation in the format normally used by the supplier (the acquirer having ensured that this format meets the documentation needs).
- Agreeing to the delivery of documentation in electronic form.
- Tailoring of standards to reduce documentation effort, or to eliminate documents.
- Considering the different documentation needs for each increment, e.g. early increments may not need the same amount of documentation as later increments, and fewer or partially complete documents may be agreed.
- The supplier providing on-line access to the design and development database for the acquirer's staff (in lieu of delivery of documentation).
- Ensuring that development tools, particularly for software, are compatible with the agreed or prescribed standards.

Gabb et al. [1991, 1992] provide further guidance on reducing documentation requirements for software projects and the use of documentation in electronic form.

Newer software development standards such as MIL-STD-498, in addition to supporting iterative development and acquisition models, are better suited to EA because they shift the emphasis away from documentation to dialogue and demonstration. Ironically, the reduced emphasis on standards may result in a need for the acquirer to increase the frequency and/or intensity of quality auditing of the supplier's development processes.

One approach which is *not* recommended, is to defer documentation (or at least delivery and review of documentation) until the final delivery of the system, i.e. at the end of the project or at least at the end of each phase.

6.3 Selecting a supplier

Selecting the supplier for an EA project will be difficult because of the flexible nature of the EA model and the typically complex and unprecedented systems involved. However, as in all projects, the objective of source selection is to identify the supplier most likely to deliver a system in a timely fashion that meets the performance requirements while providing value for money at an acceptable level of risk. The supplier must not only be able to show the ability and commitment to develop a cost effective solution, but must also exhibit the willingness and ability to work closely and cooperatively with the acquirer and users.

Assessing the ability of the supplier to provide a good technical solution is less straightforward in EA projects, because the totality of the work required is not known at contract award. As shown in the previous section, the costs of changing the supplier during the project are high, increasing the risk involved in an inadequate evaluation of the supplier's capability. One method of reducing this risk is to use a competitive first phase, with two (or even three) shortlisted tenderers expanding on their initial proposals, and demonstrating their ability to undertake the task, their understanding of it, and providing a preliminary design (including the proposed architecture) for evaluation (see section 6.2.2).

The supplier for an EA project should have the following characteristics:

- a. The ability to complete the entire system, not just the first phase, which may be less difficult to provide than later phases. Subsequent phases could require a higher level of innovation and technical skill.
- b. Demonstrated expertise and experience in the operational domain. Where many requirements are initially specified at a high level, a large degree of creativity and skill may be required. In addition, the supplier must be able to contribute to the evolution of requirements (see section 8.6) and therefore must be able to establish credibility with the users.
- c. The ability to work in a close, interactive working arrangement with the acquirers, users and other stakeholders.
- d. A demonstrated ability and willingness to make progress and risks visible, and to involve the acquirer in solving problems.
- e. A record of providing value for money in comparable projects. Once committed to a supplier the acquirer is at a disadvantage when negotiating contracts for later phases. Exploitation of an acquirer in

previous projects will increase the perceived risks of using such a supplier in EA projects.

While the past performance of a potential supplier is an essential part of supplier selection, it is often difficult to establish the reasons for poor performance in previous projects, or the parties to blame for it. To reduce the subjective nature of such information, the performance of suppliers in *all* major projects needs to be assessed by an independent review authority, both during and on completion of projects, and used as the basis for assessing past performance. Such an assessment would need to take into account the interaction between the acquirer and supplier, recognising that the capability of the acquirer is also an important factor in project success or failure.

Another difficulty in using past performance as a selection issue is that the personnel involved in a previous project may not be proposed for the current project, or a totally different part of the organisation may be involved. Similarly mergers and splitting of companies may affect the assessment. This applies equally to whether past performance was good or bad, and must be considered in source selection.

The ability of a supplier to form a close working relationship with the acquirer will also be dependent on more than the supplier's staff and experience. One additional factor is the use of subcontractors, where the relationship with subcontractors may be kept at arm's length by the prime contractor, in the interests of maintaining control. Such arrangements will also often increase the number of people who need to be involved in critical discussions, reducing the responsiveness and effectiveness of overall project management.

Another such factor is significant geographical separation between the acquirer, users and supplier, and between contributing elements of the supplier's organisation. Regardless of what measures are proposed to reduce this risk, the opportunities for and benefits of face to face meetings are likely to be reduced because of long distances between the different parties, and the ability to set up such meetings quickly will also be reduced. While technology can assist in improving informal communication at a distance (e.g. by means of telephone, electronic mail and video conferencing) such mediums are likely to be less effective in establishing and maintaining rapport between the different parties. Applying this risk factor to source selection is likely to limit the use of offshore suppliers for EA projects in Australia, for example.

6.4 Project management

6.4.1 General

No acquisition process including EA can be effective unless all the participants in the process are competent. This includes the ultimate users, the acquirers, the suppliers and the supporters [Shore 1990]. Project management is recognised to be more difficult in EA projects than in traditional approaches because of the following factors:

- a. **Multiple iterations.** Some activities need to be undertaken numerous times: contracting, test and evaluation and requirements specification, for example.
- b. **Volatile requirements.** These are evolving constantly, hence need careful management.
- c. **Interaction.** Greater interaction between stakeholders, and a better level of communication is needed.
- d. **Less time.** Decisions need to be made more quickly, necessitating more staff with higher skill levels.

A number of successful EA projects have credited a strong project management team for their success [e.g. AFCEA 1990]. Such a team will include staff who are talented, experienced, resourceful, versatile and pragmatic, who can maintain a systems view and exercise good judgment. The acquirer's project team should be as competent as the supplier's if they are to be efficient buyers [Shore 1990]. Attributes of the acquirer's team are discussed in section 6.1.3.

Early planning should specify project management requirements, identifying the numbers of staff, and the experience and skills required by staff. It may be appropriate to form the project team earlier than in other projects to plan the management strategy, to establish relationships with supplier, users and other stakeholders, and to select and prepare the project management processes and tools [DSMC 1995].

6.4.2 Interaction between the acquirer, supplier and users

A close working relationship between the acquirer, supplier and users is critical to the success of EA projects [Graham 1989; DSMC 1995; AFCEA 1993]. One method which might be considered to enhance and maintain the relationship is a 'partnering' arrangement, where all parties develop and endorse a partnering agreement, separate from the contract. The agreement should also be reinforced by regular interaction reviews, where the effectiveness of the 'partnership' is examined and proposals are made for its improvement. These should be quite separate from management and technical reviews which will normally focus on progress rather than the relationship between the different parties.

The partnering agreement should address such issues as the common objectives (primarily the delivery of a quality product) and the intent and commitment to work towards these objectives. It should also address the roles and responsibilities of the different elements of the acquirer, supplier and user teams, the interaction between them, and mechanisms for conflict resolution, including referral of problems to higher authorities. Further information on partnering is provided by DCI 2/95 [1995].

Many texts on partnering emphasise the common objectives of the parties, while minimising the essential differences in motivation and goals. It is likely

that a good working relationship will depend on all parties understanding the differences in viewpoints and motivation of other parties, and compensate for them in fostering the relationship. For this reason the agreement should also clarify the differences between parties.

The partnering agreement should be developed and agreed as early as possible after contract award (or even earlier if possible) and, as suggested above, should be reviewed regularly during the project. One method of achieving this is by an extended workshop, lasting a week or more, to develop both the partnering agreement and the detailed management plan for the project. The workshop may also be used to gain a shared understanding of the project, and to provide training to participants in forming and working in effective teams.

6.4.3 Risk management

Performance risk should be lower in EA projects due to the incremental approach, which allows problems to be found earlier and confined in scope. However, the potential for cost and schedule risks is higher, due to requirements uncertainty and the dynamic nature of the project.

Risk management should be an inherent part of the management process, and performed proactively on an ongoing basis by project staff. Risks should be continually reassessed, and the maintenance of dynamic risk lists should be encouraged and monitored. Developers may view risk differently from the acquirers who in turn may have a different perspective to the users. All perspectives have to be taken into account when making decisions based on assessment of risk.

It is a common error to overlook risks which are not directly associated with either the product or imminent milestones. In an EA project it is important to consider risks to *all* activities which may impact on successful completion of the project. These include consideration of risks such as:

- Delays in project approval.
- The lack of availability of acquirer, supplier or user staff.
- The impact of staff changes (e.g. through rotation of service positions).
- Breakdowns in communication between parties.
- Risks not being declared to other parties at the time they are identified.
- The requirements not being adequately refined at the start of an increment.
- Problems in the configuration management of requirements.
- Inadequate user feedback from fielded systems.
- Deferral of test requirements.

Finally, while the acquirer and supplier should use a common risk list to identify and manage risks, it will often be necessary for the acquirer to maintain an additional list which addresses risks in managing the project as a whole, and which considers issues such as the risk of the supplier not performing adequately, and the need to compete subsequent phases.

6.4.4 Configuration management

Configuration management is the discipline of identifying the configuration of a hardware/software system at discrete points in time throughout the project with the purpose of systematically controlling the changes to the configuration and maintaining the integrity and traceability of the configuration through the system life cycle [based on Thayer and Dorfman 1990]. Configuration management has the potential to be a serious risk area in EA projects due to the likelihood of a large number of functional baselines existing at a time. As shown in an earlier study [Gabb et al. 1991], configuration management is a serious area of risk even in traditional projects.

Although configuration management is often the responsibility of the supplier, in EA projects the requirements must be managed by both the acquirer and supplier (see section 8). In this situation it is very important that both parties have a clear understanding of the currently relevant functional baselines, and that changes to these baselines are formally approved and controlled.

Effective configuration management of the system under development (including the architecture, design, hardware components, software source code, tests and test status) is essential for maintaining control of the configurations currently under review. However, it is also a critical element in the support of the system, and in the acquirer's ability to change suppliers at some stage in the future.

7. THE SYSTEM ARCHITECTURE

The system architecture is a collection of hardware and software components and their interfaces that establish a framework for the development of a system [IEEE Std 610.12-1990]. This framework has to endure a series of incremental system upgrades during an EA project.

An architecture capable of accommodating change must be specifically designed to cater for change [Isaac and McConaughy 1994]. Generally it must be:

- **Flexible** - allowing the system to accommodate changes within the existing design or implementation.
- **Scalable** - allowing the system to meet increases in performance requirements through incremental changes in the system's capacity.
- **Extensible and maintainable** - facilitating extensions and modifications to the architecture.

Designing an architecture with these properties is crucial to the success of the project. Ould [1990] suggests basing the architecture design on those aspects of the system least likely to change. Features which are volatile or which reflect volatile requirements then need to be implemented in a way that makes them easy to change. To some extent this depends on the ability of the designers (and users in conjunction with requirements engineers) to anticipate the types of

changes which might eventuate. It also depends on the use of modern and disciplined systems design practices incorporating principles such as modularity and encapsulation that improve the maintainability and enhancement of the system.

Generally, proprietary system architectures will not be suitable for EA projects, both because this will unduly constrain the selection of hardware and software commercial products, and because of the difficulty which such architectures will present to other suppliers contributing to the project, or supporting it in service. Instead 'open' system architectures, providing standard and commonly used interfaces, protocols and operating systems, should be strongly preferred [AFCEA 1993; AFSB 1992; Arthur 1992; Fairs 1992; DSMC 1995].

The architecture must support all anticipated changes throughout the project, and hence the choice and/or design of the architecture is crucial to the project's success [DSMC 1995]. Because the architecture is based on the system requirements, it is therefore very important that the requirements are defined to a sufficient level of detail to reduce the risks in architecture design [AFSB 1992], as discussed in the next section.

8. REQUIREMENTS AND EA

8.1 The form and nature of requirements

Initially, the requirements will be defined in specifications prepared by the project sponsor, user representatives and acquirer, and which define the operational, functional and performance requirements for the project. At contract award, these will be supplemented by requirements proposed by the supplier, often in a preliminary system specification. As the design of the system progresses, many additional requirements will be created including interface requirements and the requirements for hardware and software subsystems and components, i.e. the number and detail of requirements will increase as the project progresses. (It should be noted that user interfaces are also requirements, in this context.)

In this section the word 'requirements' generally applies to the full set of requirements which exists at a particular stage of the project.

8.2 The risk of inadequate requirements

It is widely recognised that the cause of the majority of complex system failures can be attributed to requirements problems. The longer these problems go undetected the more expensive they are to fix [Davis 1993]. In many cases extra effort and more capable resources spent on requirements engineering would have produced better results.

Unfortunately there is often a reluctance to commit the necessary resources to requirements engineering because of the pressure to acquire and field a system quickly. In addition, most users and many engineers have little or no requirements engineering skills or experience, and are incapable of providing an adequate statement of requirements. The apparently natural tendency of system users to express their requirements in terms of solutions makes it difficult for them, the only ones who can identify their needs, to express them satisfactorily. Statements such as 'determining the needs is too difficult in this particular case' or 'when we see some solutions we will have a better idea of what we need' are indications of this inadequacy. In most projects, staff skilled in requirements engineering are needed to assist in capturing, validating and defining the requirements to a satisfactory level.

It is a common misconception that EA is an ideal strategy for use when the basic user needs are difficult to capture. While EA can accommodate changes in requirements, and provides early builds for users to use and comment on, inadequate requirements will increase the cost and schedule, and often prevent the fine tuning of requirements which is the strength of EA. EA must not be regarded a substitute for requirements engineering. Isaac and McConaughy [1994] state that 'compromising good requirements development practices with the intent of satisfying requirements through evolvability will only aggravate the very problem we are trying to solve'.

If EA is used with inadequate requirements, the following are likely consequences:

- The cost and schedule estimates will be underestimated, leading to numerous management problems later in the project.
- The number of iterations required to achieve a satisfactory solution will use too much of the project budget, and significantly increase the schedule. This is caused by the amount of redesign and rework needed to correct errors or omissions in requirements.
- The architecture will be put at risk. If the requirements are unclear, it may be extremely difficult to design a satisfactory architecture (see section 7).
- Inadequate early systems based on poor requirements will achieve low user acceptance, reducing the users' confidence in the success of the project, and possibly reducing user commitment and involvement.
- Major changes to correct early errors will be resisted on the basis of cost and schedule, resulting in sub-optimal designs not being changed.

In other words, EA is not only a very expensive form of requirements capture, but starting with inadequate requirements will increase the performance risk, and may threaten the success of the project as a whole.

If the requirements for an initial minimal system cannot be defined, then the project is probably not ready for full scale development. Alternative approaches which might be considered include using prototyping to develop the requirements further or delaying until requirements can be clarified.

8.3 Bounding the requirements

In an EA project, it is necessary to *bound* the requirements, i.e. limit the scope of the project, for the following reasons:

- It will normally be extremely difficult to obtain approval from funding authorities if the required capability is not reasonably well defined (how well depends on the needs of the funding authorities).
- Planning is more difficult if the project goals are not clear.
- It will be impossible to design an adequate system architecture, and to validate that design, if the broad functionality cannot be determined in advance.

Bounding the requirements means identifying the capability in terms of high level requirements, so that both the broad capability, and the overall work required to meet this capability, can be estimated. If requirements are identified during development which are outside the defined bounds, the impact of these requirements will need to be estimated, and the project will normally need to seek additional approval.

8.4 Requirements volatility

It may not be possible to specify some requirements accurately or completely, because they are inherently volatile. Some causes of volatility are as follows:

- a. **Complexity.** If the requirements for a system are complex, numerous and interrelated, it will be very difficult to completely identify all requirements.
- b. **Changing requirements.** Some systems will operate in an unpredictable environment, and the operational requirements may change during development and operation.
- c. **Using new technology.** New technology, as well as allowing new solutions to existing problems, can also introduce new requirements, as new and different ways are found to meet higher operational needs.
- d. **Unprecedented systems.** Where the capability required is new, the users will have difficulty in defining needs except at a high level.
- e. **Interoperability.** Requirements for interoperability with other systems may change as other projects evolve or protocols change. In some cases these will not be predictable.

In these cases, requirements volatility is a known risk, and should be managed as such. Requirements which are potentially volatile should be identified as early as possible, their likely variation estimated and the consequences assessed. Risk control mechanisms then need to be developed for these risks.

8.5 Requirements capture and definition

There are many techniques that can be applied to the task of capturing user requirements. Examples are structured interviews, brainstorming, prototyping and storyboarding. All need skill in requirements engineering to practice them effectively. Gause and Weinberg [1989] and Gabb and Henderson [1995] provide advice on requirements capture.

Requirements definition is the process of recording the user requirements, after they have been captured and analysed, usually by means of a specification. Writing good specifications is difficult task; common problems are mistakes, ambiguities, omissions, inconsistencies and incompleteness. Gabb and Henderson [1995] provide advice on preparing requirements specifications.

The level of specification will vary according to the users' ability to define and validate the requirement. Requirements which are well known, and which can be confidently defined, will be defined to a low (detailed) level. Others may be defined only at a high level. At the start of the project, for example, requirements for the later increments may be specified only at a high level. As the project progresses, the level of understanding will increase as users gain experience with the fielded system, and these requirements must then be specified at a lower level, acceptable to support design. It is also to be expected that some requirements will change.

Where requirements can be specified initially, they should be specified to the lowest level (highest detail) commensurate with the users' ability to define them. It should also be recognised that all requirements will need to be specified to a low level at some stage, and the earlier they are specified (assuming they may be specified accurately), the easier project planning will be. In addition, at later stages during development, the need for refinement of the requirements may result either in increments being delayed, or increase the risk of proceeding with requirements at too high a level.

In the JARS project for example (see section 5) the users will be confident about what information should be recorded for a job, because this information would be derived from the manual system JARS is to replace. Similarly, the methods for calculating costs will be known, and can be duplicated in JARS. On the other hand there will be many other requirements, such as system management functions, which cannot be captured directly from the users' experience, but which must be defined prior to the first increment. Other requirements, such as those for job planning, can be defined in a general sense prior to contract award, and refined during Phase 1.

8.6 Requirements evolution

The refinement of requirements during development, whether being actual changes or the 'fleshing out' of higher level requirements, can be referred to as requirements evolution. The process of requirements evolution will continue

throughout the development process. Changes to the requirements will include the addition of new requirements and the discarding of requirements which are no longer seen as providing value for money.

An increment cannot be confidently started until the requirements have been specified to the correct level, which will depend upon the developer's experience in the operational domain. Developers with limited experience will need more specification detail than an experienced developer. The challenge for the acquirer and supplier is to know when the specification is at the correct level. If the level is too high, there is a serious risk that the implementation will meet the written requirements, but not the users' needs. Conversely, requirements which are too detailed may inhibit innovation and constrain the developer to a solution which is far from optimal. Skilled requirements engineers, preferably with experience in EA projects, will be required throughout the project to assist in the evolution and validation of requirements.

It is important that there is a plan for the evolution of requirements throughout the project. The plan, which will change frequently, needs to take into account the current level of maturity of requirements for different functions, and ensure that the requirements for each increment are defined adequately prior to the start of that increment. In particular the plan needs to address:

- The establishment of a requirements review process and approval authority.
- The mechanisms by which feedback on requirements will be generated, collected and analysed. These should include both passive mechanisms (e.g. help desk, problem reports, suggestion scheme) and active mechanisms (e.g. user interviews, capability audits, prototyping). Note that not all suggestions for changes will originate from users - useful suggestions will also come from the acquirers, developers and testers throughout the life of the project.
- How requirements will be prioritised, including importance and urgency (how early the capability is needed).
- How the impact of requirements will be assessed, including cost, schedule constraints and special resources which may be needed.
- How tradeoffs will occur in the requirements baseline, including the removal or deferment of previously agreed requirements.

8.7 Requirements, solutions and COTS components

It is useful at this stage to examine how the requirements are evolved. Generally, the requirements will be evolved as a team effort by user, acquirer and supplier personnel. While the actual contribution of different people will depend on their responsibilities, skills and experience, the basic contributions will be as follows:

- The users bring knowledge and experience of the user domain, and feedback on the effectiveness of fielded releases.

- The acquirer will often act in a moderating role, assessing the impact that requirements may have on the project as a whole, but also providing personnel with specialised engineering and other skills, such as requirements engineering.
- The supplier will provide skills and experience in the development of complex systems (i.e. the solution domain), but is also likely to have a reasonable knowledge of the user domain.

The supplier may also provide examples of potential solutions, e.g. by the use of prototypes and/or simple applications. This allows the users to shape their requirements to enable feasible solutions [Wieringa 1996]. Similarly, because many solutions may be provided by commercial off the shelf (COTS) components, both in hardware and software, discussion between the users and supplier will help to identify requirements which enhance the use of COTS and existing components. Prototyping will be particularly relevant for requirements which cannot be met by COTS components [Waund 1995].

Early consideration of COTS components should also allow a full system engineering analysis prior to their selection. Many are now seriously questioning the value of acquirers mandating or even encouraging the use of COTS in complex systems, on many counts including their interoperability with other COTS components, frequency of upgrade, supportability and lack of critical information on their performance and test status [e.g. Waund 1995; Vigter et al. 1996].

8.8 Requirements management

As requirements evolve, they need to be managed so that there is a single authoritative definition of which requirements are currently agreed, and where they are to be implemented, i.e. in which increment. Initially there is one such 'functional baseline', but as requirements are reviewed and evolve during the project there will normally be at least 3 such baselines which need to be managed:

- The baseline for the build being used or tested;
- The baseline for the current increment; and
- The baseline(s) for the next and later increments.

Management of the requirements includes controlling changes to the requirements, including validation and authorisation of changes, and the handling of defects and suggestions for change and enhancement. Requirements management is important in all complex projects but particularly so for EA where losing control of the functional baselines may mean loss of control of the project as a whole.

Due to the complexity of the requirements management in EA projects it is essential that it be controlled by a documented systematic process [AFSB 1992], aided if possible by a requirements management (RM) tool. Without this rigour, managing the many changes likely to occur in an EA project may

quickly become an onerous and difficult, if not impossible, task. Modern RM tools include the ability to generate specifications automatically and implement traceability between requirements at different levels. The tracing features of these tools can be used to automatically identify which operational requirements are satisfied in which increments and to what extent (forward tracing), and what functionality is planned for a specific release (back tracing).

9. INCREMENTS AND RELEASES

9.1 The nature of increments

The increment is the basic building block of an EA project. An increment consists of the activities required to implement a working, demonstrable product. Increments can be considered mini-projects, each one building on the previous and progressing the project towards completion, as shown in Figure 1 (see page 8).

Ideally, user feedback from the operation of a fielded release should be available to shape the requirements for the following increment. In reality this is unlikely to be possible because each increment will usually immediately follow the completion of the previous one, and there will rarely be sufficient time to analyse suggestions or problems and incorporate changes. For this reason, operational experience and testing of a release is most likely to affect increments and phases after the ensuing increment.

The key to the EA model is flexibility. Increments can be scheduled according to the demands of specific projects. The overlap and level of concurrency between increments can be adjusted accordingly. Also, increments can be used for various purposes and in different ways that reduce the overall project risk. Some examples are given in the following sections.

9.2 Factors affecting increment selection

Increments should be defined initially by the requirements they satisfy. Allocating requirements to increments will rarely be straightforward, and needs to be based on a number of different, and sometimes conflicting factors, which are discussed below. These include operational needs, the stability of different requirements, system development demands and technology issues. Determination of the increments must be a joint activity between the acquirer, users and supplier. The JARS example (section 5) illustrates some of the factors involved in determining the functionality provided by different increments.

The allocation of requirements to increments, and even the number of increments, will not necessarily be static. As each build is produced and the requirements are progressively refined the allocation will need to be reviewed

[Erdman 1990]. It is essential to maintain a plan of the allocation of requirements to increments, including the rationale for the allocation.

Factors relevant to increment selection include the following:

- a. **Operational factors.** Some capabilities may be required sooner rather than later for operational reasons. The success of the project, as perceived by the users, will to some extent depend on the early provision of a useful capability (see section 9.5.4).
- b. **Requirements maturity and stability.** As described in section 8, some requirements will be better defined (more mature) and more stable than others. It will be necessary to allocate mature and enduring requirements to early increments, and to ensure that less mature requirements have been adequately defined prior to their particular increment being started. Consideration also needs to be given to the dependencies between requirements and how the implementation of one function may clarify the requirements of another. User feedback regarding some of the simpler functions may be required before some of the more complex functions can be developed.
- c. **Development factors.** Some requirements will need to be implemented before others for technical reasons. For example, a command support system may need office automation functions to be developed before the decision support tools which use those functions. Some requirements may need to be split, and parts or aspects allocated to different increments to provide a reasonable development strategy. Similarly, the proposed development methodology may influence the order in which functions are best developed.
- d. **Technology factors.** The implementation of some requirements may be dependent on advances in technology. In this case a plan needs to be drawn up to show how the technology is expected to mature and how this fits in with the overall plan for the implementation. Contingency plans will usually be needed to cater for the possibility that the technology will not be available when needed.
- e. **Risk factors.** It will often be sensible to implement those capabilities with higher functional risk in the early increments. This may occur, for example, where there is doubt as to whether the functionality proposed will provide the capability needed by the users, or where functions may be particularly difficult to implement. Giving users early experience of these functions may be the best way to resolve uncertainties. Any necessary refinements can be made in subsequent increments.
- f. **Other factors.** There will also be other factors which could affect the development of the system, but which may be outside the immediate control of the project. The development of related systems may necessitate that interface development is synchronised in both projects, for example.

9.3 Floating increments

Some required capabilities may have a high level of independence from the remainder of the system, i.e. they can be reasonably well specified, are relatively low risk and low priority, and could be developed at any time, and possibly by a different developer.

In the JARS example, the complex reporting functions or the development of on-line help might fall into this category. Such functions could be assigned to 'floating increments', which might be useful for a number of reasons. They could provide easily defined tasks for the developer during approval delays, for example. They could also be assigned to another supplier, if necessary (see section 6.2.3).

9.4 Concurrent development of increments

While the model presented in this paper normally assumes the sequential performance of increments, some level of concurrent development will often be necessary, not only to reduce the overall development time, but to achieve efficient utilisation of development staff. The potential for concurrent development increases where functions are largely independent of each other. In this situation different groups or even different companies may perform different increments or parts of increments simultaneously. The potential for concurrent development is limited by the management overhead incurred, particularly with the involvement of multiple suppliers.

When increments are performed concurrently, integration or synchronisation of the build(s) produced will be needed, prior to release. The risks of incompatibility and delay in any of the increment activities must be considered in the risk management strategy.

9.5 Releases

9.5.1 Non-fielding of releases

The build resulting from an increment does not have to be fielded, or even released - it can be used in a number of alternative ways. Fielding a release imposes large overheads, including packaging, testing, database conversion, training and maintenance. These will require effort from the supplier, acquirer and user, and may have schedule and cost impacts.

It may be decided, for example, that a release not be fielded, but instead be subjected to operational testing by the user community on a system test bed for example (see section 12.2). This will provide the advantages of user feedback without disrupting a large number of users for what may be for many of them a small increase in functionality. It is a useful approach, for example, when there

are risks in achieving satisfactory performance in a release. The decision on whether to field a release may be either planned from the start of the phase, or decided as the phase progresses.

In some cases the operational testing which precedes the fielding of a release may identify serious problems which need to be resolved before fielding. Under these circumstances it may be advisable either to defer the current increment under development, or to incorporate the needed corrections and/or changes in the current increment. In either case, it may be decided not to field this release.

9.5.2 Disabling of functions

A fielded release may have some functions disabled. These functions may not perform correctly, or not be fully implemented, or not be released to users for operational reasons. Alternatively, some functions may be made available only to selected users for evaluation. If the functions pass the evaluation successfully they could then be afforded wider release.

It is also possible that some functions might be found to be unreliable or incorrect in operational testing prior to release. In these cases it might be decided to proceed with the release, but with the offending functions disabled.

In the JARS example (see section 5) the existence of 'super jobs' may be hidden from users in early releases, even though the capability is partially provided. This might occur, for example, if the developer had decided to implement some aspects of super jobs in increment 1-1, to save effort at a later time and to reduce future risks, even though the full implementation is only planned for increment 2-1.

9.5.3 The first increments and first releases

The first increment will usually include the development of the system architecture and core system, and as such is likely to be the largest increment in terms of both cost and the time required for completion. While the provision of usable functionality (in user terms) may be relatively slow at this stage, early releases can serve other useful functions. The first release(s) may be planned solely for validation and operational testing by the acquirer and users and not be fielded, to ensure that the proposed solution is likely to meet the users' needs. This information would then be used to align future increments, and to determine which release should first be fielded. System releases also provide a visible indication of progress and can therefore be employed as a risk review function, particularly for the acquirer.

9.5.4 The initial fielded system

The initial fielded system is the first release fully tested and accepted for operational use. Careful thought needs to be given to the content of the initial fielded system because its success or otherwise may colour the users' attitude

for the remainder of the project. Gilb [1988] suggests delivering the 'juiciest' bits (the most impressive and useful to the user) first, the justification being that this will increase the users' commitment to the success of the project. If the fielded system has limited value to the users, or provides them with little or no additional capability beyond their existing resources, it is very unlikely to provide useful feedback to the project. It may also prejudice some users against the project because of the perceived lack of progress in providing functionality they need.

9.6 Change control during the development of increments

Changes should be kept to a minimum once an increment is under way. Even small changes can have a serious impact. Shore [1990] estimates that frequently changing requirements can increase costs by 30% unless frozen for each increment. Users need to understand that all but critical changes will usually be deferred to a later increment.

10. THE ROLE OF THE USERS

One of the main objectives of using EA is to provide systems which are closer aligned to what the users need. The user representatives, who will provide input on the operational and support needs, must therefore be seen as an essential and integral part of the process.

In traditional acquisition projects the user participation has often been less than might be expected, and restricted to specific activities, with the acquirer's staff representing the users at other times. For example, their participation may be limited to the period prior to the contract being awarded, when they contribute to the requirements and the evaluation of bids, and after the system is delivered, to assist in the operational testing of the delivered system. At various stages during implementation, they may also be consulted when design choices need to be made, and be asked to participate in reviews. The main argument for this limited participation is the belief that:

- Once the (grand design) contract is signed, it is the supplier's responsibility to build what is agreed in the contract.
- It is the acquirer's responsibility to ensure that the supplier meets the contracted requirements.
- The user requirements are well defined and do not require clarification.
- User interaction will increase the risks to the project because of unreasonable requests for changes.

In an EA project, however, the users need to participate throughout the project. Requirements definition and operational testing, rather than being activities at the start and end of the system development, occur continuously (or at least frequently), and will interact intimately with other development activities such as analysis, design and acceptance testing.

Not only do the users have a critical and continuous role to play in the project, but they must also understand that role, the principles of EA, and the overall plan for the project. While the acquirer still has the responsibility for delivering the project on behalf of the users (or project sponsors), the views and feedback from the user community will bend and shape the project direction. For this reason, they must have a clear understanding of the impact that their comments will have on the project. It will not always be possible, nor will it be sensible, to have acquirer staff 'protecting' the users from the realities of projects in all circumstances, or protecting the project from undisciplined or arbitrary user requests. The supplier's staff will often benefit considerably from informal discussions with users about requirements, potential solutions and design choices. Education of the users in these aspects will normally be the acquirer's responsibility and must be included in the project planning.

Some of the difficulties in involving users intimately in projects are that not all users have the same views, and some groups of users (who may be using only a few specific functions of the system) will have different needs than others. Obtaining a single agreed user opinion or projection may be very difficult. Some users, particularly military users with regular posting cycles, will not be available for the entirety of the project, and will need to be replaced. Users which are assisting the project on a part time basis may not feel a genuine commitment to the project, and may regard the project as a lower priority than other duties. It is also sometimes relevant to distinguish between the *user representatives*, who contribute to the project directly, and the *users in the field* who may be using the incremental releases of the system. Because requirements change in an EA project, the user representatives must be particularly careful to ensure that they understand the views of other users and represent them accurately [DSMC 1995]. Detailed written justification of requirements and the use of automated requirements management tools should help in this regard (see section 8.8).

Gause and Weinberg [1989] suggest the use of a 'user-inclusion strategy' which proposes how and when the users will participate, what types of users are needed and how they will be selected. They stress the need not only for having representation from all the significant user communities, but also in choosing the 'right' users, who will commit to and contribute in an enthusiastic, creative and responsible way.

One mechanism for moderating the views of the users, and to encourage a more uniform approach, is to form a 'user requirements team' which includes all the user representatives who will participate in the project and also includes acquirer and supplier staff with requirements engineering skills. This type of arrangement will allow a single formal user viewpoint to be expressed when needed, and will assist in resolving differences of opinion between users. It will also assist in the allocation of users to the different tasks required of them in the project.

Commitment of the users will depend to a large extent on how the involvement and feedback provided by the users is treated by the acquirer and supplier, and how closely early releases meet the needs of the users [Erdman 1990].

Commitment will also depend on the impact of workplace change which is perceived by the potential end users of a system. In some circumstances this may result in a lower level of cooperation, or downright opposition, where they are unconvinced of the need for the changes or are not kept aware of the proposed solutions and their benefits. In these cases it is important that workplace change is carefully planned and managed, and that future users are included in these change management activities, so they have a full appreciation of the impact that the proposed system will have.

Finally, the role of the users in the project should be formally agreed between the users, acquirer and supplier, in a written agreement. (This may be included in the contract, or a partnering agreement for example - see section 6.4.2.) The agreement should clearly identify the different but complementary roles of the acquirer and the users, and establish the conditions under which the users will interact with the supplier, including the rights of the users to direct the supplier (if any).

11. TEST AND EVALUATION

11.1 General

It should be noted that this paper does not assume a separate organisation responsible for T&E. Instead, testing is assumed to be undertaken by acquirer, user and supplier staff. This is a common situation in complex system acquisition for the Australian Defence Organisation. It is also likely that such an approach will improve the responsiveness of testing, and enhance the relationship between the different parties.

Systematic test and evaluation (T&E) is important in EA because the results will usually feed back quickly into the project. For the purposes of this paper, the following test and evaluation functions are identified. These will usually be relevant for each increment release.

- Supplier development testing, which may involve the acquirer or users, possibly as observers.
- Acceptance testing, where the supplier demonstrates that the performance is as agreed.
- Operational testing, where the acquirer and users validate the system, and determine its overall suitability against the users' need. This will usually be the basis for the decision as to whether to field the release or not.
- Evaluation of the system under operational conditions. This should include both the analysis of comments and problem reports from users, and proactive evaluation procedures.

Because these activities may occur with each release, the efficiency of T&E is very important in EA projects. This includes both the time required for testing and the effort required. Inefficiencies in testing will be magnified, because of the number of times they will occur. In this situation, careful T&E planning is essential.

T&E is potentially more difficult in an EA project because of the following factors:

- There will rarely be enough time to undertake a 'traditional' T&E programme for each increment.
- Decisions need to be made regarding what to test in each release, based on the functions that have been introduced, or modified, and the potential of these to affect other increments, for example.
- The results of T&E are needed as early as possible for feedback to the project.
- The distinction between developmental and operational testing may be less clear as these activities are combined.

11.2 Acceptance tests

The acceptance tests for increments need to be agreed early, prior to the start of the increment if possible, and be based on the required capability of the proposed release. The reason for this is to clearly establish both the capability that the increment will provide, and the method by which the delivery of that capability can be measured. In other words, the agreed capability of the increment should be unambiguously defined by the tests to which the release will be subjected. It will also reduce to some extent the risks perceived in the testing activities, particularly by the supplier.

11.3 Planning test and evaluation

The T&E plan needs to be specifically tailored to the project and system under development [AFCEA 1993]. Burge [1995] proposes tailoring of T&E on a risk assessment basis. Some of the factors to consider when assessing risk for a particular increment include:

- Complexity and size (for software) of the increment.
- Complexity of previous increments - increments required to interoperate with previous increments will carry a higher risk factor.
- Requirements stability.
- Deployment factors - releases scheduled for operational fielding have a higher risk factor than those that are required for demonstration purposes, for example.
- Development time.
- Maturity of the technology involved.
- The cost to the users of system failure.

- The reversibility of the installation of a release, i.e. how easy it is to revert to the previous release.

The risk associated with each increment should be estimated at the start of the project and needs to be reassessed regularly.

Specific measures which might be considered for T&E in EA projects include:

- Minimising unnecessary testing - by using carefully analysed regression testing techniques.
- Combining and coordinating acceptance and operational testing where possible to prevent duplicating testing in the two activities.
- The maximum use of automated testing to reduce the cost and time for testing, recognising that some tests will need to be repeated for each increment.
- Identifying a standard test environment (see section 12.2) which may be used for each set of tests where possible, and which may evolve with the system under development.
- Streamlining the reporting of anomalies and comments arising from testing and evaluation, and ensuring that test results are made available promptly to all relevant parties.

Involvement of users is obviously important to the success of the T&E effort. T&E planning needs to assess the extent of this involvement and show how it will be assured. Users include operational users, maintainers, trainers, support staff and any other personnel with a requirement to use the system. The T&E process must ensure that there is a mechanism for collecting, assessing and actioning user feedback.

Planning also needs to address the location of testing, and the constraints imposed by the test location and environment. Candidate locations may include the developer's premises, the acquirer's premises (possibly using a test bed) or a user site.

11.4 Correcting defects including the use of warranties

Testing and evaluation of the system, including operational use, is likely to reveal a number of defects which will need to be corrected. While some defects may require immediate correction, or may prevent the fielding or rejection of a release, the need to correct defects, and the schedule for their correction needs to be considered in the overall context of the project. If the impact of a defect is not large, or there are adequate work-arounds, deferring its correction may be a cost effective strategy. For example, it may be intended to redesign or modify the offending component in a later increment, where the defect may be superseded or where its rectification will require less effort (and be more effectively addressed) when combined with related tasks.

The development of warranties and other defect correction mechanisms should consider these aspects, and be designed to prevent the dogmatic enforcement of

contractual clauses which may in fact be against the long term interests of the users, the acquirer and the supplier.

Where defects can be tolerated for some time, their correction might be deferred to the end of the phase, when time may be available due to delays in the approval of the following phase.

12. ALTERNATIVE AND SUPPLEMENTAL ACTIVITIES TO EA

This section examines some techniques that may be used as alternatives to, or to supplement evolutionary acquisition.

12.1 The use of prototypes

Prototyping is a powerful technique that can be used to refine requirements. A prototype in this context is a working model of a system or subsystem, which emphasises specific parts of that system. Using prototypes should be considered prior to a project when the risk associated with a set of requirements is too high to commit to a development contract. Experimentation with a prototype may allow users to clarify requirements and reduce or remove this risk. Experience gained from the use of the prototype may mean that a traditional acquisition model can be used from that point onwards. Vonk [1990] and Davis [1991, 1993] provide a good introduction to prototyping. Capability and technology demonstrators are examples of prototypes.

Prototypes which look and feel like the potential system are particularly valuable when users are not familiar with the technology which is likely to be used to implement their system. They allow the users to evaluate potential solutions, which will usually result both in changes to their requirements and in better solutions. These changes can be incorporated in the prototype and validated.

An added advantage of such prototypes is that they increase user involvement and can enhance user/developer interaction. In this regard prototypes will often be beneficial when used in conjunction with an EA strategy. In the JARS project, for example (see section 5), prototypes of the job planning tools might be developed as part of Phase 1, to improve all parties' understanding of what is required. This will reduce the requirements risk for Phase 2 for these functions, and allow more confident cost estimation.

Prototyping can also be useful in analysing the capability and acceptability of COTS system components, reducing the risk in component selection.

Undisciplined use of prototyping can introduce risks, however, because prototyping is essentially a solution oriented approach, i.e. the users may focus

on a potential solution to their problem, rather than concentrate on their actual needs. The initial prototype design and limitations in the prototyping environment may have a significant influence on the way they view the system and interact with it [Lichter et al. 1994]. For these reasons, prototyping should not be seen as a tool which somehow magically captures requirements. Rather, it is a tool for the evolutionary refinement and validation of requirements initially determined by other means.

Simulation and modelling can also be used to provide a more abstract form of prototyping, where aspects of the behaviour of the system under consideration or its requirements can be examined in more detail. In this case there may be more benefit to the requirements engineers and system designers than to the users directly, who often have difficulty understanding and relating to abstract representations of the system.

12.2 The use of test beds

A test bed in this context is an environment used for testing all or part of the system. It may include complex simulation of the environment in which the system will operate, including simulation of system interfaces. In an EA project, the test bed itself will often need to evolve as the system itself evolves.

For some applications, a test bed will be necessary to undertake comprehensive development and operational testing prior to fielding. After development testing, some releases may not be fielded but tested only in the test bed environment. An example of such an application might be a ship's combat system, where the cost of fielding and the cost of system failure is likely to be high. However, the concept is equally valid for an information system, for example, where the test bed may include typical test databases and other information. It may be essential for testing systems which require a high level of trust, such as safety and security critical systems, where the penalties for system failure are high.

The test bed is likely to be useful for both the supplier, for development and acceptance testing, and for the users, for operational testing and training. It may often be cost effective to have two or more such identical facilities, possibly with one installed at the acquirer's or users' premises, to allow development testing, operational testing, and/or training to be performed concurrently, and another at the supplier's premises.

12.3 Pilot systems

A pilot system is a fully operational system (although it does not necessarily have to have the complete range of functions envisaged in the production system) developed for the purpose of gauging system performance. It may be installed only at one site, or for a selected group of users for operational use.

In an EA project, a pilot system might be used to reduce the risks associated with proceeding with development of the full system especially where a large number of vague requirements exist. Using such an approach would reduce the impact of fielding releases at a large number of sites, for example. The pilot system would form the basis of the requirements specification of the full system.

Pilot systems are suited to situations where a capability is required at multiple sites or where multiple systems are required. The system can be trialed by representative users before providing systems for multiple sites, or commencing mass production of the system. The benefits of a pilot system include reduced risk associated with fielding the system, fewer problems associated with transitioning from existing systems, and reduced training overheads because of the experience gained by users of the pilot system.

13. TROUBLESHOOTING

Despite the best efforts in planning and systems engineering, endeavouring to select the best supplier, employing sound project management practices and giving requirements the attention they deserve, things may still go wrong. In this section some possible problems are examined, and remedial actions are suggested to get the project back on track.

As in all projects, problems will often result in penalties to cost, schedule and performance, with strong interrelationships between all three. One of the difficulties in implementing remedial plans in a timely fashion is that many serious problems arise slowly, with the first obvious signs being revealed to the acquirer long after the problem should have been recognisable. Another disturbing aspect for acquirers is that too often the 'solution' to the problem is to sacrifice performance [Marciniak & Reifer 1990].

It is obvious that it is strongly in the acquirer's interests to detect problems early. To some extent EA will be better at revealing problems than traditional strategies, because of the higher level of interaction between the acquirer and supplier, the greater visibility of the development process, and the fact that system deliveries (in the form of releases) occur relatively early in the project, allowing both the users and the acquirer to get tangible indications of quality and progress.

Regardless of the suggestions below, we strongly recommend that all problems be analysed in the context of the overall project and the users' needs. EA results in a highly dynamic environment and in this situation it is too easy to be seduced into solving immediate problems and satisfying short term milestones, at the expense of the project's long term goals.

Finally, we provide no guarantees on the effectiveness of the actions suggested. Each project, acquirer, supplier and user is different, and will react differently to different stimuli. We can guarantee, however, that problems will persist while

their causes remain unchallenged, and recommend that all involved in projects try to understand and seek out the causes of problems and remove them, rather than trying to attack the symptoms. Some of the actions suggested below will be unpalatable, even unacceptable, in some projects, or to some acquirers or suppliers. In these cases, we suggest that they be at least considered in the analysis of the problem and the search for remedies. Many of actions will also result in delays as the problems and causes are analysed, and solutions are being developed. We suggest that in many cases the consequences to the project of relatively short delays will be less serious than the risks of inaction or cosmetic fixes.

13.1 Interaction and communication problems

Problem: There are problems in the interaction between the acquirer, users and supplier. These may be revealed by confrontations, personal conflicts, lack of visibility or willingness to reveal or discuss problems, high staff turnover or lack of access of acquirer staff to the development staff. This is a serious problem in any project, and may lead to project failure in EA.

Suggested action: (1) Review the reasons for and causes of the problem. Discuss these with the supplier and users, with an aim of reconciliation.

(2) Consider holding an extended professionally facilitated workshop to discuss and resolve the problems. Ensure that sufficient time is set aside to improve the probability of reaching agreement (a week may be necessary). Consider the institution, modification or reaffirmation of a partnering agreement (see section 6.4.2) and the possibility of holding more frequent face to face meetings.

(3) Identify staff who may be responsible for unnecessary conflicts, and consider (or request) their reassignment to more productive duties.

13.2 Cost blowout

Problem: The cost of the project is likely to be much higher than estimated. Typical causes include inadequate requirements, leading to frequent changes and requirements blowout (see section 13.4), or the development task has been underestimated.

Suggested action: (1) Review the requirements as they currently exist and prioritise them according to the user needs. The purpose of this is to establish a validated high level functional baseline for the remainder of the project, and identify requirements which may have marginal value.

(2) Undertake with the supplier a realistic re-assessment of the design and cost of implementation of this baseline. This information may then be used to seek further funding, to reduce the project aims, or to justify cancelling the project.

13.3 Schedule slippage

Problem: An increment falls behind schedule, and further slippages appear likely. Possible causes include unrealistic estimates and schedules, poor risk management, insufficient or inadequate development effort.

Suggested action: (1) Investigate the reasons for the slippage, and ensure that these are being addressed by the supplier.

(2) Review the project schedule including the users' needs for capability at specific times, and replan the schedule. In replanning the project, the following may be considered:

- The potential for some tasks to be carried out concurrently.
- The rescheduling of less important functions to later increments.
- The assignment of some functionality to other suppliers.

13.4 Acquirer decision making is slow

Problem: The acquirer is coming under increasing criticism for slow decision making, or is taking excessive time to review or test project deliverables, or to provide required information to the supplier. This is an indication that the acquirer may not have sufficient staff with the needed skills and experience. It is also a sign of problems to come - the usual supplier reaction to acquirer unresponsiveness is to reduce interaction and visibility and to 'work to contract', rather than to suffer the delays in involving the acquirer in decision making.

Suggested action: (1) Review the causes of the problem, including a review of the need for acquirer staffing against actual.

(2) Increase the staffing or project support to alleviate the problem, at least on a temporary basis. Replace non-performing staff.

13.5 Delays in gaining phase approval

Problem: There are significant delays in gaining approval for the next phase. This will cause problems for the supplier in maintaining the project team. Loss of supplier staff to other projects may impact the cost and schedule of the next phase.

Suggested action: (1) Review the causes of the problem - failure to gain funding approval is often an indication of lack of satisfaction by the funding authorities that risks can be controlled (see section 6). Simplifying the needs for the phase or setting less optimistic objectives may improve the likelihood of approval.

(2) Consider activities which might be used to fill the gap. In most cases interim funding will be needed. Possible measures include:

- Rectification of problems identified as warranty issues (see section 11.4).
- Reducing the next phase to one relatively small increment.
- Insertion of floating increments (see section 9.3).

13.6 Requirements blowout

Problem: The users are identifying significant requirements beyond the bounded requirements proposed for the project. These cannot be incorporated in the current phase without higher approval. The additional requirements may be an indicator that the original goals of the project have changed, or that the original requirements definition was inadequate.

Suggested action: (1) Review the new requirements, with an aim to finding out their priority and the reasons for their surfacing at this time. Consider also the relationship of the new requirements with the requirements as a whole. It may be that external events have invalidated existing requirements. Also review the possible impact of the new requirements on the cost and schedule.

(2) Depending on the significance, interaction, impact and priority of the new requirements, the following should be considered:

- Deferment of the new requirements to a follow on project.
- Incorporation of the requirements in a new or currently planned phase.
- Seeking funding approval for the project change of scope.
- Replanning the project as a whole.

13.7 Excessive requirements volatility

Problem: The number and significance of requirements changes being suggested by the users is causing serious problems, including excessive rework in each increment. This may be an indicator that the original requirements were inadequate, that the requirements evolution process is not effective, or that the supplier's design is not being responsive to the needs of the users.

Suggested action: (1) Review the reasons for the excessive volatility.

(2) If the reason is inadequate initial requirements, institute an immediate review of the requirements, including supplier staff. It is possible that the project is off course and will need replanning.

(3) If the reason is the evolution process, overhaul the process to ensure that all requirements are validated, that the users, supplier and acquirer are communicating properly, and that the supplier has a sound understanding of the users' needs.

(4) If the reason is poor system quality, review the acceptance procedures and the supplier's development methods and standards, including quality

standards. Defer the next increment if necessary to ensure the needed changes are made.

13.8 Slow requirements evolution

Problem: The requirements are not being refined quickly enough to ensure that decisions can be made in time for the planning of increments. This is usually an indicator that insufficient priority is being placed on requirements evolution by the acquirer and supplier, or inadequate user commitment. It is a sign of problems to come, as increments are based on inadequate requirements, or as increments are deferred and the requirements evolution activity is rushed.

Suggested action: (1) Review the reasons for the problem. Emphasise the importance of requirements evolution and ensure that the appropriate user, acquirer and supplier staff are made available for this activity when required. Consider the use of mechanisms to assist in requirements evolution, such as prototypes (see section 12.1).

(2) Consider advancing functionality from later increments, where the requirements are better defined.

(3) Avoid starting increments for which the requirements are insufficiently evolved.

13.9 Excessive prices for later phases and increments

Problem: The supplier's price for later increments appears excessive when compared with functions developed in earlier increments, and is assessed as not being value for money. (If the price appears high, but is still rated as reasonable value, accepting it may be the preferred option.)

Suggested action: (1) Seek more information from the supplier in justification of the apparently high price, including more detailed pricing on the package. This can be a very sensitive issue with the potential to erode trust between the acquirer and supplier, and the acquirer needs to be certain that the price is artificially inflated before proceeding. It may also be useful in some cases to undertake cost investigations or benchmarking to be satisfied that the price is reasonable.

(2) If the problem still exists, consider the following:

- Not proceeding with the function which appears to be overpriced.
- Looking for other functionality in the project which might be tendered competitively (see section 6.1.4).
- Reviewing the overall project cost estimates.

13.10 Architecture problems

Problem: The architecture is found to be inadequate to support functions needed in later increments, or to incorporate technological advances, and cannot be readily modified. This is a serious problem, as discussed in section 7. Architectural problems can be very expensive to fix.

Suggested action: (1) Seek detailed information on the nature of the problem, and the solution proposed by the supplier. Review the need for the functions and/or technology, and the impact on the users' in not incorporating them.

(2) There will often be no simple solution to this problem. If the problem is serious enough, the project as a whole could be at risk. It is essential that any solutions are considered in the context of the users' needs and the overall project. Replanning of the project may be necessary.

(3) If the situation appears serious, seek an independent assessment of the problem and possible solutions, even though this may cause a schedule slip.

13.11 Technical problems

Problem: The supplier claims serious unforeseeable technical problems, that the specified performance either cannot be achieved, or cannot be achieved within the agreed price.

Suggested action: (1) Review the requirement to gauge the effect of the reduced performance, or to validate the requirement. This may result in some modification of the requirement.

(2) Seek an independent investigation of the problem. This will give an indication of the seriousness of the problem, possible solutions, and also provide an indication of the supplier's technical competence.

(3) If the performance is achievable at a reasonable price, enforce the contract. If additional costs are involved, e.g. in further technical investigations or studies, it is preferable that these be paid to another organisation, rather than the supplier.

13.12 Deferment of functions

Problem: The supplier is showing an increasing tendency to want to defer the implementation of functions to later increments. This is normally a precursor to more serious problems such as schedule slippage or claims of technical problems, and as such is an important indicator.

Suggested action: (1) Investigate the reasons behind the deferment of functions. If due to technical or schedule problems, refer to sections 13.11 and 13.3.

(2) Functions which involve a higher performance or development risk should be addressed early in the project. Unfortunately these functions are more difficult to estimate, and may take higher quality staff and more effort to implement. Consequently they are often candidates for deferment when the project schedule is at risk. Deferring such functions is often unwise, because their resolution may impact on the design of the architecture or other functions. By agreeing to deferment, the acquirer will often expose the project to greater risk, and defer the risk's impact to a time when it may be too late to control it adequately.

13.13 Deferment of associated deliverables

Problem: The supplier wishes to defer the delivery of products (such as documentation) which are associated with, but not necessarily part of, the actual system until some time after delivery of the system. This will usually be caused by the supplier's having problems in meeting the schedule, as discussed in section 13.3.

Suggested action: (1) Review the need for the associated deliverables, and the supplier's ability to meet the schedule. In doing so, consider the value of the associated deliverables to the acquirer and/or user, and investigate why their delivery at this time was considered necessary in the project plan. Replan if necessary.

(2) Review the supplier's adherence to the agreed development methodology and standards, including quality standards. In some cases the required products may be intended to be by products of the development process. If they are being developed after the event, their quality will often be lower.

13.14 Non-delivery of agreed functions

Problem: The supplier delivers a lower level of functionality than specified in the contract. This will be detectable at development and acceptance testing, but should normally be detected much earlier (e.g. in the design documentation), if the acquirer has the appropriate visibility of development. This may be a precursor to more serious problems such as schedule slippage or claims of technical problems.

Suggested action: (1) Seek a detailed explanation for the shortfall in functionality. Depending on the reason, further actions are addressed in sections 13.3, 13.11, and 13.12.

13.15 Shortfall in quantitative performance

Problem: The release has less performance than agreed, in terms of speed, response times, data access times, number of users supported, or other

quantitative measures. This is usually an indicator of inadequate design or lack of supplier competence, and may also indicate problems with the system architecture. This will normally be revealed in development or acceptance testing.

Suggested action: (1) Seek a detailed explanation of why the shortfall occurs, and what remedial action is intended by the supplier. Review the requirements with regard to this performance.

(2) Refer to the actions suggested for the specific problem identified, e.g. technical problems (section 13.11) or architectural problems (section 13.10).

13.16 Poor reliability of fielded system

Problem: The fielded system is unreliable reducing its operational effectiveness. This may be an indicator that the system has been fielded prematurely. Problems in reliability are also likely to erode confidence in the project and reduce the effectiveness of user feedback on operational performance needed in the EA process.

Suggested action: (1) Investigate the symptoms and cause of the reliability problem. Typical causes may be inadequate maintenance, development defects (such as software defects), inadequate testing (development and/or operational), or inadequate user training (leading to incorrect or unanticipated system use).

(2) Depending on the likely cause of the problem, review one or more of the following: system support (including training), testing and development procedures.

(3) Consider deferring the fielding of new releases until the problems are resolved, and/or reverting to an earlier version of the system.

13.17 Conflicts in supplier/user access to the fielded system

Problem: The users or supplier staff are not being provided with the access they need to the operational fielded system. (In some cases this may be extended to include acquirer and support staff.) This may be caused by specific performance problems in the system (requiring additional diagnostic effort by the supplier) or inadequate planning of the interaction between the project, operational and other elements.

Suggested action: (1) Bring all relevant stakeholders together and review the operational and project objectives in the light of this problem. Place special emphasis on the consequences of lack of access of staff in each case.

(2) Review and modify the current plans and agreements with regard to access to the operational system (see sections 6.2.4 and 6.2.5).

14. CONCLUSIONS

There may be a tendency to think that EA is an easy option for acquisition, that it simplifies acquisition, or that it reduces the need for early project definition activities. None of these are true. In practice EA requires more planning, more effort and more discipline than traditional acquisition approaches if it is to be successful. We believe however that there are real benefits to be gained from the EA approach, but there are also significant risks. For some projects an iterative approach such as EA is essential.

We hope that this report has provided a deeper understanding of the benefits, penalties and risks in EA, and has showed how the benefits may be achieved, the penalties minimised, and the risks controlled.

15. REFERENCES

Ackerman, Robert K. 1995, "Rapid Development Speeds Command System Delivery", *Signal*, Vol.49, No.7, March.

AFCEA 1993, *Evolutionary acquisition study*, The Armed Forces Communications and Electronics Association, June.

AFSB 1992, *Modernisation of the Worldwide Military Command and Control System*, Air Force Studies Board, National Research Council, National Academy Press Washington DC.

Arthur, Lowell Jay 1992, *Rapid evolutionary development - Requirements, prototyping and software creation*, John Wiley & Sons, New York.

Burge, Albert R. 1995, *Test and evaluation in support of evolutionary and incremental acquisition*, White Paper, Defense Acquisition and Technology (USD(A&T)), U.S. Department of Defense, September.

CEPMAN 1992, *The capital equipment procurement manual*, Australian Department of Defence.

Davis, Alan M. 1992, "Operational prototyping: A new development approach", *IEEE Software*, Vol.9 No.5 September.

Davis, Alan M. 1993, *Software requirements: objects, functions and states*, PTR Prentice Hall, Englewood Cliffs, New Jersey.

DCI 2/95 1995, *Partnering - guidelines for project/procurement staff*, Departmental Contracts Instruction No. 2/95, Australian Department of Defence.

Dedrichsen, Loren D. 1990, "Lessons learnt from two US army evolutionary acquisition projects", *Proceedings of the EPIS '90 Symposium*, June.

DISC 1995, *Evolutionary acquisition*, Defence Industry Study Course, Syndicate Group Six, Victoria Syndicate Project.

DMD 1995, *Acquisition and Logistics 'Tiger Team' study of iterative acquisition (including Evolutionary and Progressive Acquisition)*, Defence Materiel Division, Australian Department of Defence, 16 November.

DMD 1996a, *Defence Acquisition Organisation evolutionary acquisition 'tiger' team study - Draft CEPMAN 1 Chapter on acquisition models and approaches*, Defence Materiel Division, Australian Department of Defence, 04 December.

DMD 1996b, *Defence Acquisition Organisation evolutionary acquisition 'tiger' team study - Draft CEPMAN 1 Chapter on evolutionary acquisition*, Defence Materiel Division, Australian Department of Defence, 29 November.

Downward, Bernadette G. 1994, "A brave new world: Melding systems and software engineering", *Proceedings of NCOSE*.

DSMC 1995, *Joint Logistics Commanders guidance for use of evolutionary acquisition strategy to acquire weapon systems*, The Defense Systems Management College Press, Fort Belvoir, Virginia, May.

EIA/IS-632 1994, *Systems Engineering*, EIA Interim Standard 632, The Electronics Industry Association, September 20.

Erdman, Ben 1990, "Evolutionary acquisition of command and control information systems - The Defense Communications Agency's WWMCCS experience", *Proceedings of the EPIS '90 Symposium*.

Fairs K.G. 1992, *Evolutionary acquisition of command and control systems by the Department of Defence*, Project Paper ACSC 8101, Department of Computing Science, Australian Defence Force Academy, Canberra.

Gabb, Andrew P., Peter C. Pollard, Stefan F. Landherr and Rudi J. Vernik 1991, *Tailoring DOD-STD-2167A - A survey of current usage*, Technical Note WSRL-TN-57/91, Defence Science and Technology Organisation.

Gabb, Andrew P., Peter C. Pollard, Stefan F. Landherr and Rudi J. Vernik 1992, *Recommendations for the tailoring of DOD-STD-2167A*, Report ERL-0637-RE, Defence Science and Technology Organisation.

Gabb, Andrew P. and Derek E. Henderson 1995a, *Navy specification study: Report 1 - Industry survey*, Technical Report DSTO-TR-0190, Defence Science and Technology Organisation.

Gabb, Andrew P. and Derek E. Henderson 1995b, *Navy specification study: Report 3 - Requirements and specifications*, Technical Report DSTO-TR-0192, Defence Science and Technology Organisation.

Gause, Donald C. and Gerald M. Weinberg 1989, *Exploring Requirements - Quality Before Design*, Wiley.

Gilb, Tom 1988, *Principles of software engineering management*, Addison Wesley, Wokingham, England.

Giordano, Feliciano, Bill Wong and Lee McCollum 1991, "Rapid Development Speeds Path for Command System", *Signal*, Vol.45, No.8, April.

Graham, D.R. 1989, *Incremental development: Review of nonmonolithic life-cycle development models*, Information And Software Technology, Butterworths, Guildford.

Isaac, David and Gail McConaughy 1994, "The role of architecture and evolutionary development in accommodating change", *Proceedings of NCOSE*.

Lichter, Horst, Matthias Schneider-Hufschmidt and Heinz Züllighoven 1994, "Prototyping in industrial software projects - Bridging the gap between theory and practice", *IEEE transactions on software engineering*, Vol.20, No.11, November.

Marciniak, John J. and Donald J. Reifer 1990, *Software acquisition management - Managing the acquisition of custom software systems*, John Wiley and Sons, Inc., New York.

MIL-STD-498 1994, *Software development and documentation*, Military Standard, US Department of Defence.

MIL-STD-499B 1994, *Systems engineering*, Draft Military Standard, US Department of Defence.

Ould, Martyn A. 1990., *Strategies for software engineering - The management of risk and quality*, Wiley.

Royce, W.W. 1970, "Managing the development of large software systems - Concepts and techniques", *Proceedings of WESCON 1970*.

Shore, David 1990, "Evolutionary development of C3I systems", *Proceedings of the EPIS '90 Symposium*.

Sommerville, Ian 1989, *Software engineering*, Third Edition, Addison-Wesley Publishing Company.

Thayer, Richard H. and Merlin Dorfman (ed.) 1990, *System and software requirements engineering*, Tutorial, IEEE Computer Society Press, California.

Vigter, Mark R., W. Morven Gentleman and John Dean 1996, *COTS software integration: State of the art*, Software Engineering Group, National Research Council Canada, January.

Vonk, Roland 1990, *Prototyping - The effective use of CASE technology*, Prentice Hall, Englewood Cliffs, NJ.

Waund, Carolyn K. 1995, "COTS integration and support model", *Proceedings of NCOSE*.

Wieringa, R.J. 1996, *Requirements engineering - Frameworks for understanding*, John Wiley & Sons, Chichester.

Using Evolutionary Acquisition for the Procurement of Complex Systems

Derek E. Henderson and Andrew P. Gabb

(DSTO-TR-0481)

DISTRIBUTION LIST

Number of Copies

AUSTRALIA

DEFENCE ORGANISATION

Task sponsor:

FAS Defence Materiel 1

S&T Program

Chief Defence Scientist)
FAS Science Policy) 1 shared copy
AS Science Corporate Management)

Director General Science Policy Development 1
Counsellor, Defence Science, London Doc Control sheet
Counsellor, Defence Science, Washington Doc Control sheet

Scientific Adviser to MRDC Thailand Doc Control sheet
Director General Scientific Advisers and Trials) 1 shared copy
Scientific Adviser - Policy and Command)

Navy Scientific Adviser 3 copies of Doc Control sheet
and 1 distribution list
Scientific Adviser - Army Doc Control sheet
and 1 distribution list

Air Force Scientific Adviser 1
Director Trials 1
Director Aeronautical & Maritime Research Laboratory 1

Electronics and Surveillance Research Laboratory

Director Electronics and Surveillance Research Laboratory 1

Chief Information Technology Division 1

Research Leader Command & Control and Intelligence Systems 1

Research Leader Military Computing Systems 1

Research Leader Command, Control and Communications 1

Executive Officer, Information Technology Division Doc Control sheet

Head, Information Architectures Group 1

Head, C3I Systems Engineering Group 1

Head, Information Warfare Studies Group 1

Head, Software Engineering Group 1

Head, Trusted Computer Systems Group Doc Control sheet

Head, Advanced Computer Capabilities Group Doc Control sheet

Head, Computer Systems Architecture Group Doc Control sheet

Head, Systems Simulation and Assessment Group Doc Control sheet

Head, Intelligence Systems Group Doc Control sheet

Head Command Support Systems Group 1

Head CCIS Interoperability Lab Doc Control Sheet

Head, C3I Operational Analysis Group Doc Control sheet

Head Information Management and Fusion Group Doc Control sheet

Head, Human Systems Integration Group	Doc Control sheet
Author (Mr Derek Henderson, C3ISE Group, ITD)	25
Publications and Publicity Officer, ITD	1
DSTO Library and Archives	
Library Fishermens Bend	1
Library Maribyrnong	1
Library DSTOS	2
Australian Archives	1
Library, MOD, Pyrmont	Doc Control sheet
Forces Executive	
Director General Force Development (Sea)	Doc Control sheet
Director General Force Development (Land)	Doc Control sheet
Navy	
SO (Science), Director of Naval Warfare, Maritime Headquarters Annex, Garden Island, NSW 2000.	Doc Control sheet
Assistant Chief of Naval Staff - Materiel	1
Army	
ABCA Office, G-1-34, Russell Offices, Canberra	4
Assistant Chief of the General Staff - Materiel	1
Air Force	
Assistant Chief of the Air Force - Materiel	1
S&I Program	
Deputy Director, Defence Intelligence Organisation	1
Library, Defence Signals Directorate	Doc Control sheet
Acquisition Program	
Assistant Secretary, Joint Project Management	4
Director, Information Management and Communications Engineering	1
First Assistant Secretary, Capital Equipment Programs	1
Assistant Secretary, Project Policy and Evaluation	1
Director, Command, Communications and Computing Systems	1
Director, Project Management Education and Training	1
First Assistant Secretary, Industry Involvement and Contracting	1
Assistant Chief of the Defence Force for Logistics	1
B&M Program (libraries)	
OIC, Defence Regional Library, Canberra	1
Officer in Charge, Document Exchange Centre (DEC),	1
US Defence Technical Information Center,	2
UK Defence Research Information Centre,	2
Canada Defence Scientific Information Service,	1
NZ Defence Information Centre,	1
National Library of Australia,	1
Universities and Colleges	
Australian Defence Force Academy	1
Library	1
Head of Aerospace and Mechanical Engineering	1
Deakin University, Serials Section (M list), Deakin University Library,	1
Senior Librarian, Hargrave Library, Monash University	1

Librarian, Flinders University	1
Other Organisations	
NASA (Canberra)	1
AGPS	1
State Library of South Australia	1
Parliamentary Library, South Australia	1

OUTSIDE AUSTRALIA

Abstracting and Information Organisations	
INSPEC: Acquisitions Section Institution of Electrical Engineers	1
Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts	1
Documents Librarian, The Center for Research Libraries, US	1
Information Exchange Agreement Partners	
Acquisitions Unit, Science Reference and Information Service, UK	1
Library - Exchange Desk, National Institute of Standards and Technology, US	1
SPARES	10
Total number of copies:	102

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA				1. PRIVACY MARKING/CAVEAT (OF DOCUMENT) N/A	
2. TITLE USING EVOLUTIONARY ACQUISITION FOR THE PROCUREMENT OF COMPLEX SYSTEMS			3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)		
4. AUTHOR(S) DEREK E. HENDERSON ANDREW P. GABB			5. CORPORATE AUTHOR Electronics and Surveillance Research Laboratory PO Box 1500 Salisbury SA 5108		
6a. DSTO NUMBER DSTO-TR-0481	6b. AR NUMBER AR-010-111	6c. TYPE OF REPORT TECHNICAL REPORT		7. DOCUMENT DATE MARCH 1997	
8. FILE NUMBER N9505/13/51	9. TASK NUMBER ALO95/244	10. TASK SPONSOR DMD	11. NO. OF PAGES 74	12. NO. OF REFERENCES 41	
13. DOWNGRADING/DELIMITING INSTRUCTIONS N/A			14. RELEASE AUTHORITY Chief, Information Technology Division		
15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT APPROVED FOR PUBLIC RELEASE					
16. DELIBERATE ANNOUNCEMENT NO LIMITATION					
17. CASUAL ANNOUNCEMENT NO LIMITATION					
18. DEFTEST DESCRIPTORS Evolutionary acquisition, Project planning, Procurement, Acquisition					
19. ABSTRACT This report presents the results of a study into the future use of evolutionary acquisition (EA) in the procurement of complex systems for the Australian Defence Organisation. It identifies the benefits, penalties and risks of EA projects, analyses the different activities involved, and makes recommendations for project planning, management and development activities. Its objective is to provide both an understanding of the dynamics of evolutionary acquisition projects and guidance for the use of EA in practice.					